# Effects of learning style on performance outcome in a distance education setting

Karen M. Mattison
*Rowan University*

# Effects of Learning Style on Performance Outcome in a Distance Education Setting

By:

Karen M. Mattison

A Thesis

Submitted in partial fulfillment of the requirements of the

Master of Arts Degree in Higher Education with a specialization in teaching Computer Science

The Graduate School

At

Rowan University

May 5, 2002

Approved by:

Professor

Date Approved: 5 · 20 · 02

## ABSTRACT

**Karen M. Mattison**

**Effects Of Learning Style On Performance Outcome In A Distance Education Setting**

**2001/2002**

**Dr. Jennifer S. Kay**

**Masters in Higher Education with a Specialization in Computer Science**

The purpose of this study was to determine the relationship that existed between learning style and performance outcome for a Distance Education setting. The study included three groups of students taking an introductory computer science course during the spring/02 semester at a small Community College in New Jersey. Each group was taught by the same instructor, used the same syllabus, and was expected to complete the same number and type of tests and assignments. However, the first group was a traditional lecture-based section. The second, was a traditional lecture-based section, but enhanced somewhat with a course web site interface. The last group was a non-traditional, distance education section, which completed all their course requirements via the web site interface.

The duration of the study lasted 16 weeks. Upon its conclusion the participants were given a questionnaire whose primary focus was to gather information about prior PC experience, prior programming experience, and the amount of time the student used the course web site. The results of this questionnaire coupled with individual learning style were and further enhanced with demographic information, were documented. Added to this was general progress, attendance, and participation information. A dominant learning style pattern seemed to emerge for the successful distance education student, one with a C average or better. This would lead one to conclude that a direct relationship does exist between learning style and performance outcome in a distance education setting.

## MINI-ABSTRACT

**Karen M. Mattison**

**Effects Of Learning Style On Performance Outcome In A Distance Education Setting**

**2001/2002**

**Dr. Jennifer S. Kay**

**Masters in Higher Education with a Specialization In Computer Science**

The purpose of this study was to determine the relationship that existed between individual learning style and performance outcome in a distance education setting. To determine this, student related information concerning learning style, demographics, progress, participation, and attendance was collected during an introductory computer science course. A dominant learning style pattern seemed to emerge for the successful distance education student, one with a C average or better. This dominant learning style pattern seems to establish a relationship between learning style and performance outcome in a distance education setting.

## Acknowledgments

i. I'd like to thank my husband Ken for his support and assistance throughout the preparation of this project.

ii. Also, I'd like to thank Dr. Jennifer Kay, whose help and support throughout my degree program has allowed me to realize my academic goals.

## Table Of Contents

# List of Tables

# Introduction

## *Introduction to the Study*

Distance education (DE) can be traced back to the early 1700s where its form was mostly one of correspondence, with the participants interacting via the mail. A newer form emerged in the 1960's, "Technology-based DE. This newer form had its entrance beginning with the introduction of audiovisual devices into the schools at that time. This was later enhanced by the introduction of the Internet and the capabilities it then provided" (Jeffries, 2001).

It is this latest form of DE that is so prevalent and primarily the reason behind its surge in popularity. Reasons include the technological advancements brought about by the World Wide Web that allows for the processing of text-based as well as non-text-based data (i.e, audio, video, and graphics). The information technology industry has responded with an array of instructional software applications that allow the instructor to capitalize on these new advancements and thus create an interesting on-line education experience.

One such software application that falls into this category is WebCt (www.webct.com). Some additional products worth mentioning are Blackboard (www.blackboard.com ) and Yahoo Education ( www.yahoo.com ), but WebCt seems to be in vogue at present. Each of these products gives a course designer or instructor a generic layout for a web-based course, which can then be fleshed out with specific course materials and customized to the instructor's needs and likes. This product is web-based and features a wide variety of built-in learning and teaching tools (e.g., discussion boards, chat rooms, e-mail capability) that can enhance the DE students overall experience. In addition, the product allows the course web site to be password protected, access is given only to those students who are registered and of course the instructor (CALICO Software Report, 2001).

In addition to the advancements to DE provided by the web, is the shift in age of the typical higher education student. No longer is the majority of the university population falling into the age range of 18 to 25 years of age. More often than not, the majority of the student population is older, mobile, far beyond the age of 25, and saddled with a full plate of responsibilities (Palloff, Pratt, 2001).

Adult students, regardless of age, need flexibility while pursuing your typical degree program due to personal and professional demands being placed on their time. Even the 18 – 25 year old student needs flexibility at times. During a recent commencement address, a key speaker mentioned that the average full-time student needs longer than four years to complete a typical degree program due in part to the financial needs of these students necessitating employment while attending school. Both colleges and universities, in order to capture this customer base, must respond in kind with a DE presence that includes a full array of course offerings, as well as degree programs that can be completed remotely.

***Background***

DE, while gaining in popularity, is not without its controversy. The academic community, as well as industry in general, has been slow to embrace this mode of higher education. Both educators and employers feel that DE-type courses are inferior to the more traditional lectured-based course offerings, thereby degrading the whole learning experience for the student (Galusha, 2001).

In order to influence the skeptic's attitude towards DE in a positive way, careful attention must be paid to the quality of the DE course material being offered to the higher education student. Curricula and assessment materials must be developed that rival that of the traditional classroom. In addition, carefully planned comparisons must take place periodically between the classroom setting and the "cyber" setting to validate what format/technology of DE works best, for which courses, and for which students. If any quality-related discrepancies exist, they should be resolved immediately. This should help to encourage the DE skeptics to embrace this mode of higher education and all the benefits it can provide to the adult student.

***Learning Style's Significance***

For many years now educators have noticed that some students prefer certain methods of learning to others. These methods include:

- More or less instructor interaction
- Independent vs. group learning activities

- More or less technology used in the delivery of the course

These learning preferences form a student's unique learning style and have aided instructors in the planning of course related activities (Cartnal, Diaz, 1999).

Monitoring a DE course for its performance outcome by studying the influence that individual learning style plays on overall performance would prove useful in the continued evaluation of DE. It is well known that all modes of teaching, traditional as well as DE benefit some learners in some way while posing problems for others. Therefore, it would seem important to explore how learning style might affect performance outcome for the DE student (deLeon, Killian, 2000).

## ***Study details***

In order to capture the learning style pattern for the successful DE student, one with a C average or better, a test would need to be administered. The test itself would be given only once, since it is unlikely that an adult student will sway very much from their individual learning styles. In addition, the test should focus on specifics concerning the learning style behavior of the DE student. Obtaining proper measurements for these individual learning styles will aid in the profile development of the successful DE student.

There is an array of learning style assessment tools that exist today. Any one of these tools could aid in the facilitation of such a DE study. It is important however, to choose one that targets and generates feedback about the learning behaviors of this type of student. Therefore a learning style test of this nature should focus on those items necessary for a DE student's success. These items would include:

- Attitude toward school
- Motivation level
- Managing time effectively
- How tense they become with academic tasks
- Concentration on school-related tasks
- Organization of new material
- Main idea selection
- Study aid preparation
- Self testing strategy
- Test taking strategy

The data gathered from such a test would be graded, analyzed for individual learning style patterns, and returned to the student with comments about their scores and how they might improve their success rate in future DE courses. Both the student and their instructor need to be aware of what a students learning preferences are and if they show the potential to be a successful DE candidate. When a DE course begins, a frank discussion, virtual if necessary, concerning individual learning styles should be encouraged. This discussion should help to identifying which dominant patterns of learning style seems to emerge for the class and what accommodations might be made to the array of course-related activities so that all participants could enhance their performance.

An overabundance of research on learning styles and the role it plays in DE does not exist at present. Most of the studies thus far focus on the comparison of DE with that of its lecture-based counterpart for achievement level outcomes (i.e. drop rate, completion rate, and final grades). Proponents of DE contend that the findings thus far prove only that distance learning is "as good as" traditional education if conducted properly. (The Institute for Higher Education, 1999).

Future research then in this area should focus on these key points:

- How best to conduct DE courses properly
- How best to use DE technology in the delivery of those courses
- How to determine what technology works best in the delivery of a DE course
- How to establish a DE student profile

In order to generate significant results for a study of this type, certain procedures should be established for the sake of consistency. First, it should occur on at least two groups of students taking the same course and with the same instructor. Next, The first group's mode of learning should be traditional. The second group's mode of learning should be DE. Third, the students would not be able to switch between groups after the study began. Fourth, the actual sample used for the study would be chosen from each group of students randomly, following the close of registration. One final note, it might prove useful to have a third sample group included in this study, one that would complete its course work traditionally, but enhanced somewhat with a DE web interface.

In addition to the above details, each student will be required to complete their course work using the same syllabus, textbooks, tests, assignments, and technology. Last, all students, regardless of their mode of learning, will be expected to complete the same number and type of activities as their counterparts and with in the same time frame. Progress, attendance, and participation level will determine final performance outcomes for each study participant. Added to this data will be the students learning style data and demographic information. Those students who emerge with a C average or better in each group will be further analyzed. A dominant learning style pattern should emerge in each setting. This pattern should be different for each mode of learning and should aid in the development of a successful DE student profile.

***Problems to be studied***

- What differences exist in learning style among the participants in a DE course with those in a more traditional setting?

- Does a dominant learning style pattern emerge for the successful DE student?

## Review of the Literature

Stevenson (2001) reports that many institutions today are experimenting with computer-based forms of DE. This mode of learning is known by many terms that include distance learning, remote learning, open education, web-based learning, and computer-mediated DE. Regardless of its title, there are two broad categories that make up this field of study. The first: asynchronous, includes most forms of DE that allow the delivery of the course material via an instructional web site where the students are not communicating with the instructor at the same time. Here, all the lessons and their associated assignments are first posted to the course web site, the student then view the course material and complete the assignments using standard mail, e-mail, audio, video, and discussion boards whenever convenient for them.

The second, synchronous, has instructors and participants, although geographically remote, interacting in real-time using chat rooms, conference calls, and/or video conferencing in order to complete course related activities. It is important to note that there could easily be a third category that might include some asynchronous and synchronous components. For example, the students and instructor might communicate by traditional mail, or e-mail asynchronously during one lesson and use conference calling during the second. Clearly a combination approach to DE instruction could then be realized.

Palloff and Pratt (2001) have suggested that while providing an alternative to the lecture-based approach, DE must be monitored closely for quality. Critical concerns have begun to arise: technological infrastructure, copyright issues, choice of software interfaces, and accreditation issues that effect the faculty, the students, and administration alike. Current research suggests that any additional studies in this field should focus on an array of issues, in particular are those issues concerning student performance measurements.

Gold and Maitland (1999) state that a large amount of material exist that is devoted to DE research including policy papers, "how to" articles, and essays. In addition, there is, though not an insignificant body of, original research. With few exceptions, the bulk of these writings lead one to believe that the learning outcomes of students who participate in conventional classroom instruction are similar to those students using web-based instruction, if the said instruction is adequately conducted. However, individual learning preferences could play a key role in the success of the student in the DE setting. Therefore, a frequent look at DE compared to its counterpart must take place. Close attention must be paid to the individual learning preferences and how this data can then be incorporated back into the material to enhance future performance.

Diaz and Cartnal (2000) conducted related research that suggests a student's performance might rely heavily upon his or her own learning style preferences. It was further suggested that students might self-select into or away from DE courses due to their personal learning style preferences. As a result, success of the DE student, and/or a DE program, may ultimately depend on the DE student's learning style, the instructor's handling of this data, and the instructor's ability to incorporate the individual learning styles back into the course activities and assessment strategy, ultimately influencing performance outcome.

Galusha (2000) seems to support the study by Diaz and Cartnal (2000), but goes further to suggest that DE is student-centered learning and the instructor needs to be well aware of how each student learns best before proceeding with a course. Thus, knowing the demographic data and learning style of each DE student could identify the barriers to their learning. Although knowing the students' characteristics and needs may not always guarantee success in a DE course or program, this information would allow its defense as contributing to student success. Additionally, knowledge about student characteristics and internal/external student motivators will help to us understand who is likely to participate in DE, who is likely to end up successful, and why others choose not to participate.

Dill and Mezack (1991) in some earlier research completed a study on the relationship of learning style to performance outcome. They found that learning style did play a key role in individual performance, but that certain demographic data might also prove significant in predicting success or non-success in a DE setting. This could include GPA, credit-hours-completed, age, gender, and marital status.

Carlson (2001) seems to support Dill and Meszack's (1991) research but goes further in suggesting that women make up the majority of DE learners. His belief is that women in general faced many challenges such as, childcare issues, work schedule conflicts, in addition to their learning style that prevent them from achieving a positive performance outcome. In addition, he suggests that a woman's reason for choosing DE might simply be to allow for flexibility while completing their higher education pursuits.

Becker and Dwyer (1998), in yet another related study, determined that individual learning preferences were one of the aspects of a students personality that appears to play a role in determining how much a student learns, but does not elaborate of the specifics of those learning preferences.

Some fields, such as engineering, are already implementing learning style assessments in the classroom so that further study of individual learning style and any relationships it may have with a student's performance can be realized. As an example, Felder (1996) cites the use of learning style measurements as one means of improving the match between instructors' methods and a student learning style. His research goes further to suggest that there are many dimensions of a student's learning style and that and this can include either a preference for learning verbally or one for learning visually. Computer technology, found extensively in DE, relies heavily on visual stimuli and therefore may appeal more to students who prefer to learn visually.

Luk (1998) added to the available DE research by conducting a study of nursing students taking DE course work, these studies were similar in nature to Becker and Dwyer's work (1998). Luk determined from his research that a positive relationship did exist between field-independence in a DE setting and academic achievement of a student. Field-independence and field-dependence are learning styles that an individual can possess. The field-independent learner is basically a self-starter, motivated, and finds that the impersonal nature of DE is an ideal learning environment. Thus, placing heavy emphasis on specific analytical skills and providing little opportunity for interaction. The field-dependent learner is the complete opposite. This type of student thrives in the more traditional lecture-based setting.

Ito and Sumrall (1993) had completed some early research that supports Luk's (1998) study. They compared the learning style of two groups of students taking a language course. One group was taught at a distance and one group was not. High levels of achievement were determined to be driven by a student's motivation level, learning strategies used, gender, and number of credits completed. The results of the study suggests that motivation and learning style were primary determinants of an individual success in school. Gender, although not a primary determinant, was an additional factor in determining a students overall performance outcome.

## *Time line & Summary*

1991 – Dill and Mezack

- Demographic data and learning preference determined a student's performance outcome, but mode of learning was not specified.

1996 – Felder

- Learning preferences might determine a student's best choice of instructor, ultimately leading to a positive performance outcome.

1998 – Ito and Smrall

- Motivation, learning strategy, and number of credits completed were determining factors for a positive performance outcome. However mode of learning was not specified.

1998 – Luk

- Field independent learning is a trait of the successful DE student.

1998 – Becker and Dwyer

- Learning preference plays a role in how much a student learns.

1999 – Gold and Maitland

- DE, if adequately conducted, has a similar learning outcome as the more traditional mode of learning if conducted properly.

2000 – Gulusha

- Instructors can be more effective in a DE setting if they know the learning style of their students.

2000 – Diez and Cartral

- Learning style preference determines the best mode of learning and leads to a successful performance outcome overall.

2000 Carlson

- Suggests that DE learners are mostly women.
- Choose DE for its flexibility.

2001 – Palloff and Pratt

- DE would benefit from quality assurance measures such as :
    - Technology infrastructure
    - Copyright issues
    - Software interfaces
    - Accrediting issues

2001 – Gold and Maitland

- Determined that DE consisted of two broad categories:
    - Asynchronous
    - Synchronous

In summary, although there have been numerous studies concerning DE students, there was not enough documentation that addressed the impact an individual's learning style plays on their success in a DE course or what the profile of that student would be. In addition, no standards were mentioned that specifically identify the type of test needed to determine the DE student learning style, the name given to each learning style, scale for measuring these learning styles, or mix of technology to deliver course material. This has lead to inconclusive evidence in determining the relationship that exists between learning style and performance outcome in a DE setting.

DE students must not only navigate their way through the course material, but also an array of DE related technology as well as additional software that might be included for course related activities. Therefore, it is the position of this paper to perform a study comparing a group of DE students with a group of traditional students studying and introductory Computer Science course. The goal, to see the effects that learning style have on performance outcome in each setting. The array of technology used in each setting, the test used to capture the learning style of the study participants, and what learning style behaviors along with behavior measurement data will be documented.

## Research Methodology

For this study a stratified sampling technique was used on the population, which consisted of all students currently registered for Introduction to C++ during the spring 2002 semester at Salem Community College. Initially three sections of Introduction to C++ were planned for the spring semester. The first was a high school advanced placement section, which was taught using the traditional lecture-based approach, and with no accesses to the Internet course web site (Group 1). The second section was an evening, adult student's section that would be taught in the traditional fashion (Group 2). The last section was a DE section that worked with the instructor remotely, via the Internet 90% of the time, to complete course related activities (Group 3).

Regardless of the section, all students were taught by the same instructor, used the same textbook and syllabi, and complete the same number and type of assignments. Each section was then asked to participate in the study by signing a release form. The form identified explicitly who conducted the research and what data would be collected. In addition, each participant was advised that they would remain anonymous in the final document preparation. The release form (see Appendix A) along with instructor comments was returned to the student at the end of the study so that the student would clearly understand their learning style.

The learning style assessment tool was administered to each study participant. The actual test, Learning and Study Strategy Inventory (LASSI) (see Appendix B) was chosen for several reasons. First the LASSI was available in both an Internet format as well as a paper and pencil format, which would work out well for the DE student who participated remotely. Second, the actual test kit was inexpensive to purchase per student, keeping study costs down. Third, the actual test took less than 30 minutes to administer. Finally, the test itself addressed noteworthy items of interest about DE students such as:

- Attitudes (ATT)
    - How clear is the student about his or her own education goals?
- Motivation (MOT)
    - Does the student stay up-to-date with class assignments?
- Time Management (TMT)
    - Is the student well organized?
- Anxiety
    - Does the students worrying interfere with concentration?
- Concentration
    - Is the student easily distracted?
- Information Processing
    - Can the student apply new information to past expediences?
- Selecting Main Ideas
    - Can the student focus on the key points of a lecture?

- Self Testing
    - Does the student review before a test?
- Test Strategies
    - Does the student prepare appropriately for test?

Once the data was collected from the assessment, it would eventually be analyzed for mode of learning, motivation level, time management, concentration, anxiety, test taking strategies, selection of main ideas, information processing, study aid preparation, and self testing strategies. Measurements in each of these areas, and for each student, would be charted along with performance level, assessment scores, and attendance. Demographic data such as Age, Gender, Program of Study, and GPA would be added also.

Throughout the duration of the study, which lasted approximately one sixteen week period, the primary technology used by the student participants was an array of Microsoft software that included:

- Microsoft Visual C++
- Microsoft Internet Explorer or Netscape Navigator
- Microsoft Windows 2000
- Microsoft Outlook Express
- Microsoft Office

One additional piece to the technology puzzle was the WebCt interface, but that was only used by the DE section and part-time by the Enhanced section.

As the course unfolded, both the DE section and the Enhanced section needed to complete textbook readings, discussion board participation requirements, and assignments using the WebCt interface (see Appendix C) and the MS Visual Studio and/or MS Office software. Any student assessment was administered using the WebCt interface, graded and returned back to the student via the mail. All assignment exchanges between the student and the instructor occurred using e-mail. These assignments were graded by the instructor and returned back to the student in the same fashion as well.

The non-DE sections completed the same number of textbook readings, and programming assignments. In addition, they used the same array of Microsoft software to complete those assignments as their non-DE and Enhanced-DE counterparts. However, discussion was completed in class, as well as many of the programming assignments. Their tests, a paper and pencil version, were completed in class as well. They interacted with the instructor on a regular basis.

Upon completion of the semester the data gathered concerning performance outcome was analyzed together with both individual learning style and demographic data. This was then charted using Microsoft Excel. The data was sorted in ascending order first by mode of learning, then by final grade (see Appendix D). The successful student, those with a C average or better, appeared to show a dominant learning style pattern. This dominant learning style pattern was similar for each successful student within his or her respective group. In addition, this learning style pattern was different for each mode of learning. Thus the role that learning style plays in determining performance outcome in a DE setting was realized and was further supported by demographic data as well as a particular course delivery technology.

## Analysis of the data

### *Introduction*

Over the past five months, attempts have been made to determine if individual learning style plays a key role in performance outcome in a DE setting. Being a Computer Science instructor at a small community college in southern New Jersey, I was able to implement this study concerning learning style, relatively easily. First, a popular learning style assessment tool was chosen for this study (Learning And Study Strategy Inventory (LASSI) ). It included a web interface so that those DE students who chose to participate in this study could do so. Second, a popular Computer Science course was chosen, Introduction to C++, primarily because of its popularity and ability to provided a large enough sample of students and during a given semester. Last, since the college had recently adopted the WebCt product as a standard for any future DE course development, it was chosen as the DE course web-development tool.

During the earlier stages of my research and prior to the actual start of the DE course, an effort was made to familiarize myself with the WebCt product and any course delivery issues that might arise during the semester:

- Being able to view the WebCt data on various workstations
- Being able to view the WebCt data on a workstation using different hardware and software versions that were not the same as the instructors

- Being able to view the WebCt data on a workstation that did not have the same manufacturer/version of browser software
- Being able to view the WebCt data using various Internet access facilities determining the speed needed to successfully view the WebCt course data without unacceptable response time delays.

The actual DE course that was offered to the students was based on earlier course research, which included syllabus, lesson plans, assignments, student assessment strategy, example programs, presentations, and virtual discussions.

Satisfied with the layers of technology evolved in the course's delivery, procedures were then developed for course interaction. These procedures were then tested thoroughly to ensure for a smooth delivery of the course-related material. The following items were addressed:

- Navigation procedures for the student to access the WebCT product via the school web site.
- Sign on procedures for the new user as well as the returning user
- New course addition procedures.
- Assignment completion, packaging, and delivery procedures.
- Procedures for the implementation of all programming assignments using Visual C++.
- Test taking and submission procedures.

One last item to be resolved was the student questionnaire given as part of the last lesson (see Appendix E) this was then used to gatherer information about:

- Prior programming experience
- Level of programming experience where applicable
- Prior PC experience
- Level of PC experience

So, in the spring of 2002, the DE study began and lasted for the duration of the semester. The students in each of the three sections of Introduction to C++ were asked to participate in a study by signing a release form. Those that signed the form were deemed study participants and given the learning style assessment tool in the beginning of the semester and a questionnaire at the end. This data, along with their progress and demographic information was used to determine the outcome of the study.

During the student registration process, the students self-selected into their respective sections based on age, schedule constraints, and/or interest. The number of study participants in each group was determined from the original number of registered students. The DE section ended up with seven students, the non-DE, adult section ended up with twelve students. The third section was a restricted, non-DE section of fifteen high school students, plus one adult.

With the course content resolved, the course delivery issues tested, learning style assessment strategy chosen, the course interaction procedures distributed, and the sample groups of students in place, the semester began in early September without incident. However, during the first week of the semester, each DE student was contacted and given their list of course related procedures. Added to this were instructions on how to purchase the textbook via the Internet if necessary, how and when to communicate with the instructor if questions or concerns should arise, how to transmit assignments for grading, and so forth. At this point several of the students became overwhelmed with the amount of technology they needed to become proficient with even before they opened their textbook. Some decided to drop the course due to its complexity, or simply switch into one of the more traditional sections being offered.

At the beginning of the second week there were four remaining students in the DE section, twelve in the adult, non-DE section, and fifteen in the juvenile, non-DE section. The adult in this section stopped attending and was never heard from again.

It was also during this same second week that another phenomenon occurred with the original adult section of Introduction to C++. The non-DE, adult section wanted flexibility in completing their course requirements. I then gave these students access to the web-based version of the course, allowing for a web-enhanced experience for those occasions when attending class was impossible due to personal or professional demands on their time. After exposure to the course web site, one-third of the original students ceased to attend class, but continued to complete their course work as DE students.

The study continued on from this point without further incident with the three sample groups of students. The first group was the juvenile, traditional, group of fifteen students (Group 1). The second was the adult, web-enhanced group of nine students (Group2). The third was the DE group of the original four students remaining after the 1st week's pullout, plus the four additional students (Group3). These students started out classified as non-DE, but became reclassified as DE due to continuing with the course in a DE mode. No further adjustments to the three sample groups of participants occurred from this point forward.

***Analysis***

The LASSI scales and score interpretations start this section, followed by each group's individual data:

| | |
|---|---|
| **Attitude (ATT)** | High scores are good, low scores indicate that the student needs to work on goal setting. |
| **Motivation (MOT)** | High scores are good, low scores indicate that the student needs to accept responsibilities for specific tasks. |
| **Time Management (TMT)** | High scores are good, low scores indicate that the student needs a Personal Data Assistant. |

| | |
|---|---|
| **Anxiety (ANX)** | High scores are bad, the lower the better. Indicates how well the student handles anxiety related to school. |
| **Concentration (CON)** | High scores are good, they indicate the student is effective at focusing at the task at hand. |
| **Information Processing (INP)** | High scores are good, low scores indicate the student needs instruction in methods that can be used to help organization of their thoughts. |
| **Selecting Main Ideas (SMI)** | High scores are good, low scores in a problem with this task. |
| **Study Aid Preparation (STA)** | High scores indicate the student performs this and the low score indicates they should learn how. |
| **Self Testing (SFT)** | High scores are good, and a low score indicates a need by the study to initiate some self testing strategy. |

| | |
|---|---|
| **Testing Strategies (TST)** | High scores are good and low scores indicate the student needs to learn to review for their tests. |

The following table lists the scores in each category that the average student achieved:

**T-1 LASSI average scores for each category**

| | |
|---|---|
| Attitude – 32 | Motivation – 31 |
| Time Management – 2 | Testing Strategies – 30 |
| Anxiety – 26 | Concentration – 25 |
| Information Processing – 26 | Selection of Main Idea – 18 |
| Study Aid Preparation – 24 | Self Testing – 25 |

Group1, according to the questionnaire data, had no prior programming experience and for the most part had intermediate PC experience before registering for this course. In addition, they had no prior knowledge of the course web site and were not exposed to that site throughout the duration of the semester. The demographic makeup associated with this group was 50% female, 33% minority (African American), and 100% having 0 credits completed prior to this semester. In addition, all were younger than 17 and held a GPA that exceeded the grade of 89. The additional progress and learning style data for those students who choose to participate in the study is summarized in the following tables:

**T-2 Questionnaire Data summarization for Group 1**

| | |
|---|---|
| Prior programming experience | 0% |
| Programmer experience level | n/a |
| Prior PC experience | 100% |
| PC experience level | beginner |
| % on course web site | 0% |

**T-3 LASSI data, Progress data, & Demographic data for Group 1**

| | *Student 1* | *Student 2* | *Student 3* | *Student 4* | *Student 5* | *Student 6* | *Student 7* | *Student 8* | *Student 9* |
|---|---|---|---|---|---|---|---|---|---|
| Attitude | 32 | 34 | 29 | 37 | 33 | 28 | 35 | 21 | 30 |
| Motivation | 24 | 35 | 31 | 31 | 32 | 34 | 37 | 26 | 39 |
| Time Mgr. | 24 | 22 | 20 | 15 | 21 | 28 | 30 | 27 | 37 |
| Anxiety | 6 | 25 | 23 | 20 | 20 | 23 | 22 | 38 | 16 |
| Concentration | 17 | 27 | 25 | 15 | 22 | 32 | 28 | 26 | 28 |
| Reasoning | 23 | 21 | 24 | 29 | 34 | 31 | 28 | 21 | 36 |
| Main Idea Selection | 14 | 21 | 16 | 14 | 26 | 12 | 23 | 21 | 15 |
| Study Aid Preparation | 20 | 16 | 16 | 21 | 29 | 22 | 27 | 16 | 33 |
| Self Testing | 17 | 29 | 22 | 21 | 34 | 24 | 31 | 15 | 28 |
| Test Strategies | 24 | 26 | 24 | 21 | 24 | 24 | 33 | 36 | 26 |
| Gender | female | female | female | female | female | male | male | male | male |
| Age | 15 | 15 | 15 | 16 | 16 | 15 | 15 | 15 | 16 |
| GPA | >89 | >89 | >89 | >89 | >89 | >89 | >89 | >89 | >89 |
| Program of study | Non degree seeking | Non degree seeking | Non degree seeking | Non degree seeking | Non degree seeking | Non degree seeking | Non degree seeking | Non degree seeking | Non degree seeking |
| Total credits | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Classes missed | 0 | 0 | -2 | -3 | 0 | -1 | 0 | 0 | -3 |
| Final Grade | 87 | 80 | 77 | 88 | 86 | 87 | 77 | 65 | 79 |

The top-performers of this group: student1, student4, student5, and student6 represent slightly more than one-fourth of the group's total number of students. They can be further sub-divided as follows: three females and one is male. The learning style data for Group1's high participants is summarized in the following table:

**T-4 Summarization of LASSI data for high performers in Group 1**

| NonDistance | Student 1 | Student 4 | Student 5 | Student 6 | Average of Non-DE Scores | Highest Non-DE score | Low Score |
|---|---|---|---|---|---|---|---|
| Attention | 32 | 37 | 33 | 28 | 33 | 37 | 28 |
| Motivation | 24 | 31 | 32 | 34 | 30 | 34 | 24 |
| Time Management | 24 | 15 | 21 | 28 | 22 | 28 | 15 |
| Anxiety | 6 | 20 | 20 | 23 | 17 | 23 | 6 |
| Concentration | 17 | 15 | 22 | 32 | 22 | 32 | 15 |
| Info. Process. | 23 | 29 | 34 | 31 | 29 | 34 | 23 |
| Selecting of Main Idea | 14 | 14 | 26 | 12 | 17 | 26 | 12 |
| Study Aid Prep. | 20 | 21 | 29 | 22 | 23 | 29 | 20 |
| Self Testing | 17 | 21 | 34 | 24 | 24 | 34 | 17 |
| Testing Strategy | 24 | 21 | 24 | 24 | 23 | 24 | 21 |

Group2, according to the questionnaire data, had prior programming experience and for the most part had intermediate PC experience before registering for this course. In addition, they used the course web site about 25% to 50% of the time to take tests, participate in virtual discussions, review syllabus, and obtain copies of assignments when not able to attend. The demographic makeup associated with this group was 60% female, 10% minority (African American), and 100% having completed less than 50 credit hours, prior to the start of this semester. In addition, most of their GPA scores exceeded 3.5. The learning style data and progress data, for those students who participated in the study, is summarized in the following tables:

## T-5 Questionnaire Data summarization for Group 2

| | |
|---|---|
| Prior programming experience | 33% |
| Programmer experience level | novice |
| Prior PC experience | 100% |
| PC experience level | intermediate |
| % on course web site | 25% - 50% |
| Items used on site | Discussion board, Testing, Syllabus, Course Outline |

## T-6 LASSI data, Progress data, & Demographic data for Group 2

| Enhanced DE | Student 1 | Student 2 | Student 3 | Student 4 | Student 5 | Student 6 | Student 7 |
|---|---|---|---|---|---|---|---|
| Attitude | 25 | 30 | 34 | 33 | 32 | 39 | 37 |
| Motivation | 24 | 27 | 26 | 25 | 25 | 38 | 31 |
| Time Mgr. | 22 | 19 | 21 | 18 | 17 | 33 | 27 |
| Anxiety | 29 | 18 | 29 | 26 | 36 | 36 | 28 |
| Concentration | 10 | 25 | 25 | 26 | 25 | 38 | 37 |
| Reasoning | 28 | 29 | 26 | 15 | 31 | 37 | 28 |
| Main Idea Selection | 14 | 22 | 23 | 18 | 17 | 25 | 21 |
| Study Aid Preparation | 24 | 15 | 21 | 24 | 17 | 27 | 15 |
| Self Testing | 20 | 14 | 22 | 24 | 23 | 31 | 27 |
| Test Strategies | 25 | 34 | 38 | 34 | 33 | 38 | 36 |
| Gender | female | female | female | female | male | male | Male |
| Age | 30 | 29 | 24 | 45 | 22 | 37 | 15 |
| GPA | 2.5 | 3.5 | 0 | 3 | 3 | 4 | 4 |
| Program of study | Computer Programming | Enrichment | Enrichment | Network Computer Systems | Liberal Arts | Enrichment | Internet applications developer |
| Total credits | 10 | 6 | 0 | 0 | 50 | 4 | 0 |
| Classes missed | -8 | -1 | -1 | -3 | -5 | -2 | 0 |
| Final Grade | 65 | 83 | 82 | 80 | 79 | 92 | 99 |

The top-performers of this group: student3, student6, and student7, represent one-fourth of the group's total number of students. They are then further divided as follows: one male and two females. Learning style data for Group2's participants is summarized is the following table:

**T-7 Summarization of LASSI data for high performers in Group 2**

| **Enhanced DE** | *Student 3* | *Student 6* | *Student 7* | *Average score for this group* | *High score For this group* | *Low score for this group* |
|---|---|---|---|---|---|---|
| Attitude | 34 | 39 | 37 | 37 | 39 | 34 |
| Motivation | 26 | 38 | 31 | 32 | 38 | 26 |
| Time Management | 21 | 33 | 27 | 27 | 33 | 21 |
| Anxiety | 29 | 36 | 28 | 31 | 36 | 28 |
| Concentration | 25 | 38 | 37 | 33 | 38 | 25 |
| Reasoning | 26 | 37 | 28 | 30 | 37 | 26 |
| Main Idea Selection | 23 | 25 | 21 | 23 | 25 | 21 |
| Study Aid Preparation | 21 | 27 | 15 | 21 | 27 | 15 |
| Self Testing | 22 | 31 | 27 | 27 | 31 | 22 |
| Test Strategies | 38 | 38 | 36 | 37 | 38 | 36 |

Group3, according to the questionnaire data, had prior programming experience and for the most part had intermediate PC experience before registering for this course. In addition, they used the course web site 100% of the time to take tests, participate in virtual discussions, review syllabus, and obtain copies of assignments. The demographic makeup associated with this group was 40% female, 10% minority (African American), and the majority of the group having completed less than 50 credits prior to the start of this semester. The learning style data and progress data, for those students who choose to participate in the study, is summarized in the following tables:

## T-8 Questionnaire Data summarization for Group 3

| | |
|---|---|
| Prior programming experience | 90% |
| Programmer experience level | novice |
| Prior PC experience | 100% |
| PC experience level | Intermediate |
| % on course web site | 100% |
| Items used on site | Discussion board, Testing, Syllabus, & Course Outline |

## T-9 LASSI data, Progress data, & Demographic data for Group 3

| **DistanceEd** | *Student 1* | *Student 2* | *Student 3* | *Student 4* | *Student 5* | *Student 6* | *Student 7* |
|---|---|---|---|---|---|---|---|
| Attitude | 34 | 38 | 33 | 26 | 38 | 34 | 38 |
| Motivation | 32 | 34 | 24 | 29 | 35 | 34 | 38 |
| Time Management | 27 | 33 | 17 | 18 | 29 | 16 | 33 |
| Anxiety | 25 | 32 | 25 | 20 | 25 | 18 | 28 |
| Concentration | 24 | 37 | 26 | 25 | 32 | 24 | 31 |
| Reasoning | 33 | 27 | 25 | 30 | 26 | 25 | 29 |
| Main Idea Selection | 18 | 24 | 19 | 13 | 22 | 20 | 21 |
| Study Aid Preparation | 20 | 17 | 21 | 24 | 23 | 30 | 21 |
| Self Testing | 25 | 20 | 24 | 26 | 28 | 30 | 25 |
| Test Strategies | 29 | 37 | 28 | 29 | 31 | 27 | 30 |
| *Gender* | female | male | female | male | male | female | male |
| *Age* | 51 | 34 | 42 | 37 | 32 | 29 | 21 |
| *GPA* | 3.7 | 0 | 2.5 | 3.8 | 2.5 | 3.9 | 3.7 |
| *Program of study* | Network Computer Systems | Business Administration | Enrichment | Enrichment | Liberal Arts | Network Computer Systems | Network Computer Systems |
| *Total credits* | 18 | 12 | 151 | 20 | 115 | 18 | 12 |
| *Classes missed* | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| ***Final Grade*** | W | I | W | 98 | 85 | 82 | W |

The top-performers of this group: student4, student5, and student6, represent slightly less than one-half of the group's total number of students. They are then further divided as follows: one female and two males. Learning style data for Group3's participants is summarized in the following table:

**T-10 Summarization of LASSI data for high performers in Group 3**

| **DistanceEd** | *Student 4* | *Student 5* | *Student 6* | *Average for this group* | *High score for this group* | *Low score for this group* |
|---|---|---|---|---|---|---|
| Attitude | 26 | 38 | 34 | 33 | 38 | 26 |
| Motivation | 29 | 35 | 34 | 33 | 35 | 29 |
| Time Management | 18 | 29 | 16 | 21 | 29 | 16 |
| Anxiety | 20 | 25 | 18 | 21 | 25 | 18 |
| Concentration | 25 | 32 | 24 | 27 | 32 | 24 |
| Reasoning | 30 | 26 | 25 | 27 | 30 | 25 |
| Main Idea Selection | 13 | 22 | 20 | 18 | 22 | 13 |
| Study Aid Preparation | 24 | 23 | 30 | 26 | 30 | 23 |
| Self Testing | 26 | 28 | 30 | 28 | 30 | 26 |
| Test Strategies | 29 | 31 | 27 | 29 | 31 | 27 |

The learning style averages for each of the three groups was then summarized and the following tables represent that data. Added to this is the standard deviation for each average grouping:

**T – 11 Summarization of LASSI data for all three groups**

| *CATEGORY* | *DE* | *Non DE* | *Enhanced* | *Stdev* |
|---|---|---|---|---|
| Attitude | 33 | 33 | 37 | 2 |
| Motivation | 33 | 30 | 32 | 2 |
| Time Management | 21 | 22 | 27 | 3 |
| Anxiety | 21 | 17 | 31 | 7 |

| | | | | |
|---|---|---|---|---|
| Concentration | 27 | 22 | 33 | 6 |
| Reasoning | 27 | 29 | 30 | 2 |
| Main Idea Selection | 18 | 17 | 23 | 3 |
| Study Aid Preparation | 26 | 23 | 21 | 3 |
| Self Testing | 28 | 24 | 27 | 2 |
| Test Strategies | 29 | 23 | 37 | 7 |

The high performers in each group, those with a C average or better, were identified and the following data is a summary of their demographic information:

**T-12 High Performers' demographic data for Group 1, Group 2, & Group 3**

- 5 males, 5 females
- age range was between 15 - 37
- GPA range was between 2.5 - 4.0
- Programs of study included: Enrichment, Liberal Arts, & Networked Computer Systems

A final analysis summarizes the high performers learning style data , for each group, in table T-13. This is followed by a summary of the low performers learning style data in table T-14.

## T- 13 Table represents all high performers LASSI data

| *CATEGORY* | HighAvg DE | HighAvg Non-DE | HighAvg Enhanced | StdDev |
|---|---|---|---|---|
| Attitude | 33 | 31 | 34 | 1.6 |
| Motivation | 33 | 32 | 29 | 2.2 |
| Time Mgr. | 21 | 25 | 23 | 2.0 |
| Anxiety | 21 | 21 | 29 | 4.4 |
| Concentration | 27 | 24 | 29 | 2.4 |
| Reasoning | 27 | 27 | 28 | 0.3 |
| Main Idea Selection | 18 | 18 | 21 | 1.6 |
| Support Tech. | 26 | 22 | 20 | 2.9 |
| Self Testing | 28 | 25 | 24 | 2.4 |
| Test Strategies | 29 | 26 | 36 | 4.7 |

## T-14 Table that represents all low performers LASSI data

| *CATEGORY* | LowAvgDE | LowAvgNonDE | LowAvgEnhabced | stdDev |
|---|---|---|---|---|
| Attitude | 29 | 21 | 25 | 3.8 |
| Motivation | 26 | 26 | 24 | 1.1 |
| Time Mgr. | 22 | 27 | 22 | 2.9 |
| Anxiety | 22 | 38 | 29 | 8.0 |
| Concentration | 24 | 26 | 10 | 8.6 |
| Reasoning | 23 | 21 | 28 | 3.6 |
| Main Idea Selection | 16 | 21 | 14 | 3.6 |
| Support Tech. | 16 | 16 | 24 | 4.7 |
| Self Testing | 19 | 15 | 20 | 2.6 |
| Test Strategies | 25 | 36 | 25 | 6.4 |

## Summary, Conclusions and Recommendations

After a careful analysis of the data for each group, the only learning style categories that stands out as uniquely different are the DE scores in the use of support techniques and materials category and the DE scores in the concentration and attention to academic tasks category. The remaining learning style data was just too similar to be included in developing the profile of the successful DE student.

The support techniques and materials category measures the student's ability to use or create study aids that support and increase individuals learning and retention. Using and creating study aids provides the student with the ability to create their own aids when attempting any course related assessment. It would improve both effectiveness and efficiency in learning, each a very important determinant in a successful performance outcome for a DE student.

Concentration and Attention to academic tasks allows the learning experience to be more meaningful for the student. This area tries to gauge how well the student organizes their thoughts and applies these thoughts to new material. Using our prior knowledge to help make new information understood is key to an individuals success in education. A low score in this area implies that DE students simply need more time to acquire and recall new material. Again, the DE student, distant from the instructor has this time, and this is a primary reason for a DE student choosing this mode of learning.

The demographic information collected for Group 1, Group 2, and Group3 did not seem to show any significant results or patterns. However, the data collected concerning all high-performers and the data collected concerning all low-performers seems to show that the high-performers have the following learning style preferences:

- High reasoning skill level to process new information
- High self-testing skill level for review and exam preparation
- Low anxiety towards assignments
- Positive attitude towards school
- Highly motivated

Therefore, the profile of the successful DE student, based on the data gathered, would include at least the following items:

- High learning style score for support techniques and material.
- Low learning style score for concentration & attention to academic tasks.
- Prior PC experience
- Prior Programming experience

In addition, based on the data gathered about high-performers, the initial list of learning style preferences for the successful DE student might also include the following as well:

- High reasoning skill level to process new information
- High self-testing skill level for review and exam preparation
- Low anxiety towards assignments
- Positive attitude towards school
- Highly motivated

## References

Becker, D. & Dwyer, M. (1998). The impact of student verbal/visual learning style preference on implementing groupware in the classroom. , Journal of Asynchronous Learning Networks Vol. 2, Issue 2 - Sept 1998.

Computer Assisted Language Instruction Consortium Software Report (2001). Which web course management system is right for me? A comparison of WebCT 3.1 and Blackboard 5.0. Retrieved 11/02/01, from http://astro.temple.edu/~jburston/CALICO/review/webct-bb00.htm

Carlson J., (2001). Scholars debate the future of online distance education, The Chronicles of Higher Education, Retrieved 11/07/01, from http://www.chronicle.com/distance

DeLeon, L. , Killian, J. (2000). Comparing modes of delivery: Classroom and on-line (and other) learning. Journal of Public Affairs Education, Vol. 06.

Diaz, D. P. & Cartnal, R. B. (1999). Student learning styles in two classes: Online distance learning and equivalent on-campus, College Teaching, Vol. 47, (pp. 130-135).

Felder T. & Richard, M. (1996). Matters of style (pg. 18), ASEE Prism Magazine, December, Vol. 6, No. 4.

Galusha, J. (2001). Barriers to Learning in Distance Education, Retrieved 10/25/01, from http://www.infrastruction.com/barriers.htm.

Gold L., Maitland, C., (1999). What's the difference? A review of contemporary research on the effectiveness of distance learning in higher education. The Institute for Higher Education.

Haught, P., Hill, L., Walls, R., Nardi, A. (1998). Improved learning and study strategies inventory (LASSI) and academic performance: The impact of feedback on freshmen. Journal of the First-Year Experience, 1998, Vol. 10, No. 2, pp.25 – 40

Ito, S. & Sumrall, M. (1993). Learning a language by satellite television: What Influences student achievement?( pp 31 - 48), Pergamon Press, Vol. 21, No. 1.

Jeffries, M. (2001). Indiana partnership for statewide education - Research in Distance Education, Retrieved 10/25/01, from http://www.ihets.org/consortium/ipse/fdhandbook/resrch.html .

King, J. (1996). The missing component in distance education: "Wetware" development. 1996 University of North Texas, USA, P54 - 56.

Luk, S. (1998). The Relationship between cognitive style and academic achievement. British Journal of Educational Technology (pp 137 - 147), Vol. 29, No 2.

Mezack, M. & Dille, B. (1991). Identifying predictors of high risk among community college telecourse students, The American Journal of Distance Education, Vol. 5, No. 1.

Palloff, R., Pratt, K. (2001). Lessons from the Cyberspace Classroom: The Realities of Online Teaching. San Francisco, CA, Jossey-Bass.

Rekkendal, T. (1994). Research In distance education – past, present, and future. Retrieved 11/01/01, from http://www.nettskolen.com/forskning/29/infoseek.htm.

Stevenson, N. (2001). Distance Learning Online, Foster City, CA., IDG Books Worldwide

## Appendix A

Thursday, April 04, 2002

Office of Government Grants/Sponsored Projects
Internal Review Board, Chairperson
Rowan University
201 Mullica Hill Road
Glassboro, New Jersey 08028

To Whom It May Concern:

I am writing to you today, as per my conversation on Tuesday, February 26 with Stephanie Kirby, to ask for two slight changes to my original IRB document submitted in preparation for my thesis project this past fall.

1) I would like to include students from another section of my CS121 Introduction to C++ class. These students are under 18 years of age. This change is necessary to increase my research sample, thus enhancing the results. The number of current students in the distance education group would not lead to any significant data and by increasing this number, the results would be view as more significant.

2) I would like to add a final survey to give to all of my students at the end of the semester. This survey would be anonymous, and would ask students for their opinions of the different components of the course, allowing the student to elaborate on their experiences with the distance education format and any benefit to them that it then provided.

You specified that it would be necessary for me to elaborate on the details and provide an additional consent form that would be use for the under age students, discussed above. The additional consent form follows as an attachment to this letter.

As was identified in my original IRB document, I will be pursuing research that will compare learning styles between two educational settings, the traditional and distance education. Each participant will complete a learning assessment test. The focus of the test will be to identify academic behaviors concerning: Motivation, Time Management, Concentration, Anxiety, Test Taking Strategies, Selection of Main Ideas, Information Processing, Study Aid Preparation, and Self-Testing Strategies. This test is optional and will only be administered to students who are registered for Introduction to C++ during the spring/02 semester at Salem Community College. In addition to the learning style assessment test, the instructor, the same in each setting, will collect course performance data for each student participant and merge this with their learning style. The distance education sample will additionally be asked to write a one-page summary identifying how this type of setting benefited them as a student.

Sincerely,

Karen M. Mattison

(Graduate Student completing thesis requirements for M.A. in Higher Education with an academic teaching specialization in Computer Science)

# INSTITUTIONAL REVIEW BOARD DISPOSITION FORM

| | |
|---|---|
| Karen M. Mattison | ______________________ |
| (Principal Investigator) | (Co-Principal Investigator (if applicable) |
| 460 Hollywood Avenue | ______________________ |
| | (Address of Co-Principal Investigator) |
| Carneys Point, New Jersey 08069 | ______________________ |
| | (City, State, and Zip Code of Co-Principal Investigator) |
| (856) 299-2100, mattison@salemcc.org | ______________________ |
| | (Telephone # Fax # e-mail address of Co-Prinvipal Investigator) |

TITLE OF RESEARCH Effects of learning style on performance outcomes in a distance education setting.

ADMINISTRATIVE DISPOSITION - DO NOT WRITE BELOW THIS LINE

Your claim for exemption for the research study identified above has been reviewed. The action taken is indicated below:

__________ APPROVED FOR EXEMPTION AS CLAIMED: CATEGORY # __________
Note: Anything that materially changes the exempt status of this study must be present4ed to the IRB for approval before the changes are implemented. Such modifications should be sent to the IRB Office at the address above.

__________ APPROVED FOR EXEMPTION - BUT NOT AS CLAIMED. Your claim for exemption does not fit the criteria for exemption designated in your proposal. However, the study does meet the criteria for exemption under CATEGORY # __________.

__________ A determination regarding the exempt status of this study cannot be made at this time. Additional information is required.

__________ Your proposal does not meet the criteria for exemption, and a full review will be provided by the IRB>

EXPEDITED REVIEW: __________ Approved __________ Denied __________

FULL Review: __________ Approved __________ Approved with modifications __________ Denied __________

DENIED: ______________________________________________

See attached Committee Action Letter for additional comments.

| | |
|---|---|
| ______________________ | ______________________ |
| Chair, IRB | Co-Chair, IRB |
| Date__________________ | Date__________________ |

# INSTITUTIONAL REVIEW BOARD APPLICATION FOR REVIEW OF RESEARCH

1. Type of approval review requested (check one): Full Review _______ Expedited Review __X_
   Review Exemption _____

2. PRINVIPAL INVESTGATOR: Karen M. Mattison

3. DEPARTMENT: Graduate student, M.A. in Higher Education with an academic teaching specialization in Computer Science

4. TITLE OF RESEARCH: Effects of learning style on performance outcomes in a distance education setting

5. CO-INVESTIGATORS: NONE

6. PURPOSE OF RESEARCH (INDEPENDENT PROJECT, MASTER'S THESIS, ETC.): The purpose of this project is to fulfill the requirements of a M.A. in Higher Education with an academic teaching specialization in Computer Science

7. IF YOU ARE A STUDENT RESEARCHER PLEASE PROVIDE THE FOLLOWING:

   MAILING ADDRESS: 809 Vista Way, Deptford New Jersey 08096
   EMAIL: mattison@salemcc.org TELEPHONE NO.: (856) 228-4365

   FACULTY SPONSOR NAME: Jennifer S. Kay Associate Professor, Computer Science Department, Rowan University

   PHONE: (856) 256-4593 FAX: (856) 256-4741 EMAIL: kay@elvis.rowan.edu
   FACULTY SPONSOR SIGNATURE: ______________________________
   DATE: ____________________

8. HAS THIS RESEARCH PROJECT BEEN CONSIDERED PREVIOUSLY BY THE IRB? YES OR NO
   IF YES, GIVE DATE OF LAST REVIEW: __________________

9. SOURCE OF FUNDING (IF APPLICABLE):
   _________ SBR GRANT
   _________ UNIVERSITY GRANTS (INCLUDING FOUNDATION)
   _________ CAREER DEVELOPMENT GRANT
   _________ EXTRAMURAL FUNDS
   PLEASE INDICATE AGENCY NAME: ______________________________

10. ARE YOU WORKING WITH A RESEARCHER FROM ANOTHER INSTITUTION? IF SO, BE AWARE THAT YOUR CO-INVESTIGATOR MUST ALSO SUBMIT YOUR JOINT PROPOSAL TO THE IRB AT THE INSTITUTION THAT EMPLOYS HIM/HER. ______YES __X__ NO

11. DOES YOUR RESEARCH INVOLVE ANY OF THE FOLLOWING (CHECK ALL THAT APPLY)?

| | |
|---|---|
| _____ | minors _____prisoners _____pregnant women |
| __X__ | use of the investigators' current students as subjects. |
| _____ | drugs or other controlled substances. |
| _____ | psychological or physiological stress above the level of normal everyday activities. |
| _____ | misleading or deceiving subjects about any aspect or purpose of the research. |
| _____ | collection of information which deals with sensitive aspects of the behavior (e.g., illegal activity, drug or alcohol use, sexual behavior). |
| _____ | collection of information which would place subjects at risk of criminal or civil liability if it became known. |
| _____ | collection of information which could affect subjects' financial standing, employability, or reputation if it became known. |
| _____ | examination of existing data, documents, or specimens that are <u>not</u> part of the public record. |
| _____ | children involved in your research without sensitive information about themselves or their families. |
| _____ | collecting or studying existing data, documents, records, pathological specimens or diagnostic specimens, which are publicly available and from which participants cannot be identified by anyone other than the investigator(s). |

12. WHAT IS THE OBJECTIVE OF THE RESEARCH?
*The objective of this research will be to find what relationship exists between learning style and performance outcome for a distance education setting. This then could lead to the development of a profile for the successful distance education student. Those students not meeting this profile would therefore be the weaker counterpart and might benefit from additional advisement by the instructor. In addition, the instructor could use the information gathered concerning learning profile to enhance the course so that it can accommodate a learning style variety.*

13. DESCRIBE THE DESIGN OF THE RESEARCH INCLUDING WHAT WILL BE REQUIRED OF SUBJECTS (ATTACH ADDITIONAL SHEET IF NECESSARY):
*Each research participant will be required to complete a learning assessment test that will identify academic behaviors concerning: Motivation, Time Management, Concentration, Anxiety, Test Taking Strategies, Selection of Main Ideas, Information Processing, Study Aid Preparation, and Self Testing Strategies. Of course this is optional and will only be administered to adult students who are registered for Introduction to C++ during the spring/02 semester at Salem Community College. Approximately one-third to one-half of these registered students will be completing the course requirements in an asynchronous distance education setting and will be considered the distance education sample. Those not included in the distance education sample will be the counterpart sample group and will be completing their requirements using the traditional lectured-based approach. The instructor, course duration, syllabi, lesson-plans, lab activities, homework assignments, and assessment strategy will be the same for both groups. Once the learning style tool is administered, the study will progress with data being collected by the instructor that will record each student's participation level, progress data, and attendance. At the end of the study, learning style data and performance data will be merged for each student. It is expected that those students*

*deemed successful, having a C average or better, in each sample will show a dominant learning style pattern and this pattern should be different for both groups.*

14. UNDER WHICH OF THE FOLLOWING CATAGORIES ARE YOU APPLYING FOR EXEMPTION?

___ 1.Research conducted in established or commonly accepted educational settings, involving normal educational practices such as, (i) research on regular and special educational instructional strategies, or (ii) research on the effectiveness of the comparison among instructional techniques, curricula, or classroom management methods.

___ 2. Research involving the use of social sciences or educational tests (cognitive, diagnostic, aptitude, achievement), survey procedures, interview procedures, or observation of public behavior where (I) information is not obtained in such away that the participants can be identified directly or indirectly or (ii) the participants' responses, if they became known, could not place the participant at risk of criminal or civil liability or be damaging to the participants'; financial standing, reputation, or employability. (All research involving survey and interview procedures is exempt when the participants are elected or appointed public officials or candidates for public office. However, confidentiality must be maintained when required by federal statute).

___ 3. Research involving the collection of study of existing data, documents, records, pathological specimens, or diagnostic specimens, if these sources are publicly available or if the information is recorded by the investigator in such a manner that participants cannot be identified.

___ 4. Research and demonstration projects which are funded by a federal agency and determined to be exempt by the agency head and which are designed to study, evaluate, or otherwise examine: (I) public benefit or service programs; (ii) procedures for obtaining benefits or services under those programs; (iii) possible changes in or alternatives to those programs or procedures; or (iv) possible changes in methods or levels or payment for benefits or services under those programs.

___ 5. Exemption for collection or study of existing data: research involving collection or study of existing data, documents, records, if these data are non-identifiable and publicly available or information is recorded by the investigator in such a manner that subjects cannot be identified directly through identifiers linked to the subject (codes linking names to data are considered indirect identifiers).

___ 6. Exemption for study of the department of health and human services: unless specifically required by the statute, research and demonstration projects which are conducted by or subject to the approval of the Department of Health and Human Services, and which are designed to study, evaluate, or otherwise examine:

_____ (a) programs under the Social Security Act or other public benefit or services programs
_____ (b) procedures for obtaining benefits or services under those programs;
_____ (c) possible changes in or alternatives to those programs or procedures;
_____ (d) possible changes in methods or levels of payment for benefits or services under those programs.

IF YOUR RESEARCH IS GIVEN EXEMPTION STATUS, THE FOLLOWING MUST BE STATED ON A COVER LETTER (ON DEPARTMENTAL LETTERHEAD) ACCOMPANYING ANY SURVEY OR QUESTIONNAIRE:

1. A statement that all participation is voluntary
2. A statement that you are conduction research and the reason for it (e.g., master's thesis, publication, etc.)

3. Purpose of the research - what you are investigating
4. A statement that all responses will be kept anonymous and confidential
5. A statement that participants need not respond to all questions
6. If participants are you own students, a statement that class standing will not be affected in any way based on participation
7. The name and telephone number of the Principal Investigator (PI) and faculty sponsor (if applicable)

**CLAIMS FOR EXEMPTION MAY NOT BE MADE FOR (A) RESEARCH INVOLVING CHILDREN, (B) AIDS-RELATED RESEARCH, (C) RESEARCH INVOLVING SUBSTANCE OR CHILD ABUSE OR (D) RESEARCH TO BE CONDUCTED AT THE V.A. (RESEARCH UNDER THESE CATEGORIES IS SUBJECT TO SPECIAL FEDERAL GUIDELINES.)**

**COMPLETE THE FOLLOWING ADDITIONAL QUESTIONS FOR A FULL IRB REVIEW**

15. DESCRIBE THE SUBJECTS WHO WILL BE PARTICIPATING (NUMBER, AGE, GENDER, ETC):

    ______________________________________________

    ______________________________________________

16. HOW WILL SUBJECTS BE RECRUITED? IF STUDENTS, WILL THEY BE SOLICITED FROM CLASS?

    ______________________________________________

    ______________________________________________

17. WHAT RISKS TO SUBJECTS (PHYSIOLOGICAL AND/OR PSYCHOLOGICAL) ARE INVOLVED IN THE RESEARCH?

    ______________________________________________

    ______________________________________________

18. IS DECEPTION INVOLVED IN THE RESEARCH? IF SO, WHAT IS IT AND WHY WILL IT BE USED?

    ______________________________________________

    ______________________________________________

19. WHAT INFORMATION WILL BE GIVEN TO THE SUBJECTS AFTER THEIR PARTICIPATION? IF DECEPTION IS USED, IT MUST BE DESCLOSED AFTER PARTICIPATION.

    ______________________________________________

    ______________________________________________

20. HOW WILL CONFIDENTIALITY BE MAINTAINED? WHO WILL KNOW THE IDENTIFGY FO THE SUBJECTS? IF A PRE AND POSTTEST DESIGN IS USED, HJOW WILL THE SUBJECTS BE IDENTIFIED?

    ______________________________________________

    ______________________________________________

21. HOW WILL THE DATA BE RECORDED AND STORED? WHO WILL HAVE ACCESS TO THE DATA? ALL DATA MUST BE KEPT BY THE PRINCIPAL INVESTIGATOR FOR A MINIMUM OF THREE YEARS.

    ______________________________________________

    ______________________________________________

CONSENT FORM FOR RESEARCH PROJECT for "Effects of learning style on performance outcomes in a distance education setting"

I ________________________________________ agree to participate in a study entitled "Effects of learning style on performance outcomes in a distance education setting", which is being conducted by Karen M. Mattison, currently pursuing a Masters in Higher Education at Rowan University. The purpose of this study is to evaluate the role that individual learning style plays in performance outcome, when a student is taking a distance education course. The data collected in this study will consist of individual learning style preferences, progress, attendance, and participation levels and will be submitted as a part of a Thesis documents requirements.

I understand that I will be required to take a thirty-minute test that will assess my learning style and that this test will be evaluated, recorded, and then returned with comments.

I understand that my responses will be anonymous and that all the data gathered will be confidential. I agree that any information obtained from this study may be used in any way thought best for education provided that I am in no way identified and my name is not used.

I understand that my participation does not imply employment with the state of New Jersey, Rowan University, the principal investigator, or any other project facilitator.

If I have any questions or problems concerning my participation in this study I may contact Karen M. Mattison or Dr. Stephen Waldow at (856) 299-2100.

| | |
|---|---|
| ________________________________ | ______________________ |
| (Signature of Participant) | (Current Date) |
| ________________________________ | ______________________ |
| (Signature of Investigator) | (Current Date) |

**CONSENT FORM FOR RESEARCH PROJECT for "Effects of learning style on performance outcomes in a distance education setting"**

I ______________________________ agree to participate in a study entitled "Effects of learning style on performance outcomes in a distance education setting", which is being conducted by Karen M. Mattison, currently pursuing a Masters in Higher Education at Rowan University. Due to my age I am also specifying that ________________________________, my parent/legal guardian, also agrees to allow me to participate. The purpose of this study is to evaluate the role that individual learning style plays in performance outcome for both a distance education setting and the traditional counterpart. The data collected in this study will consist of individual learning style preferences, progress, attendance, and participation levels and will be submitted as a part of a Thesis documents requirements.

I understand that I will be required to take a thirty-minute test that will assess my learning style and that this test is free, will be evaluated, recorded, and then returned with comments to each participant.

I understand that my responses will be anonymous and that all the data gathered will be confidential. I agree that any information obtained from this study may be used in any way thought best for education provided that I am in no way identified and my name is not used.

I understand that my participation does not imply employment with the state of New Jersey, Rowan University, the principal investigator, or any other project facilitator.

If I have any questions or problems concerning my participation in this study I may contact Karen M. Mattison or Dr. Stephen Waldow at (856) 299-2100.

| | |
|---|---|
| ______________________________ | ________________________ |
| (Signature of Participant) | (Current Date) |
| ______________________________ | ________________________ |
| (Signature of Parent or Guardian) | (Current Date) |
| ______________________________ | ________________________ |
| (Signature of Investigator) | (Current Date) |

201 Mullica Hill Road - Bole Annex - Phone : 856.256.4057 - Fax : 856.256.4425

**Rowan University's Office of Government Grants and Sponsored Projects**

# Fax

| | | | |
|---|---|---|---|
| To: | Professor Kay | From: | Stephanie Kirk |
| Fax: | 4741 | Pages: | 2 |
| Phone: | 4593 | Date: | 12/12/01 |
| Re: | Karen Mattison | CC: | |

☐ Urgent ☐ For Review ☐ Please Comment ☐ Please Reply ☐ Please Recycle

● Comments:

Here's a copy of the original mailed out early Nov.

Thanks

# INSTITUTIONAL REVIEW BOARD DISPOSITION FORM

| | |
|---|---|
| Karen M. Mattison | ____________ |
| (Principal Investigator) | (Co-Principal Investigator (if applicable) |
| 460 Hollywood Avenue | ____________ |
| | (Address of Co-Principal Investigator) |
| Carneys Point, New Jersey 08069 | ____________ |
| | (City, State, and Zip Code of Co-Principal Investigator) |
| (856) 299-2100, mattison@salemcc.org | ____________ |
| | (Telephone # Fax # e-mail address of Co-Prinvipal Investigator) |

TITLE OF RESEARCH Effects of learning style on performance outcomes in a distance education setting.

ADMINISTRATIVE DISPOSITION - DO NOT WRITE BELOW THIS LINE

Your claim for exemption for the research study identified above has been reviewed. The action taken is indicated below:

_______ APPROVED FOR EXEMPTION AS CLAIMED: CATEGORY # _______
Note: Anything that materially changes the exempt status of this study must be present4ed to the IRB for approval before the changes are implemented. Such modifications should be sent to the IRB Office at the address above.

_______ APPROVED FOR EXEMPTION - BUT NOT AS CLAIMED. Your claim for exemption does not fit the criteria for exemption designated in your proposal. However, the study does meet the criteria for exemption under CATEGORY # _______.

_______ A determination regarding the exempt status of this study cannot be made at this time. Additional information is required.

_______ Your proposal does not meet the criteria for exemption, and a full review will be provided by the IRB>

EXPEDITED REVIEW: K [✓ Approved] _______ Denied _______

FULL Review: _______ Approved _______ Approved with modifications _______ Denied _______

DENIED: _______

See attached Committee Action Letter for additional comments.

[illegible]
Chair, IRB
Date 11/10/01

[illegible]
Co-Chair, IRB
Date 11/9/01

## Appendix B

# LASSI

# Learning And Study Strategies Inventory

©1987, H&H Publishing Company, Inc.
1231 Kapp Drive
Clearwater, Florida 33765-2116

by
**Claire E. Weinstein, Ph.D., David R. Palmer, Ph.D.**
Department of Educational Psychology, University of Texas at Austin
**Ann C. Schulte, Ph.D.**
University of North Carolina

## Directions

The Learning and Study Strategies Inventory (LASSI) is designed to gather information about learning and study practices and attitudes. On the two forms at right, which you pull out to begin the LASSI, you will find 77 statements related to learning and studying. You are to read each statement and then mark a response according to the following key:

- Not at all typical of me
- Not very typical of me
- Somewhat typical of me
- Fairly typical of me
- Very much typical of me

To help you decide which responses to mark, we would like to explain what is meant by each term.

By **Not at all typical of me**, we do not necessarily mean that the statement would never describe you, but that it would be true of you only in rare instances. Mark an **a** for this response.

By **Not very typical of me**, we mean that the statement generally would not be true of you. Mark a **b** for this response.

By **Somewhat typical of me**, we mean that the statement would be true of you about half the time. Mark a **c** for this response.

By **Fairly typical of me**, we mean that the statement would generally be true of you. Mark a **d** for this response.

By **Very much typical of me**, we do not necessarily mean that the statement would always describe you, but that it would be true of you almost all the time. Mark an **e** for this response.

Please completely darken the appropriate letter. For example, darken the **d** if you feel that the statement is fairly typical of you.

a b c ■ e

Try to rate yourself according to *how well the statement describes you*, not in terms of how you think you should be or what others do. There are no right or wrong answers to these statements. Please work as quickly as you can without being careless and *please complete all the items.*

**Both of the forms at right, along with the Directions booklet are two-part, carbonless forms. Take care *not* to stack any of the forms on top of the other when writing since it would damage the forms below.**

**After reading the directions, tear out *both* two-part forms at right and set this booklet aside. The forms contain the statements you will respond to. This booklet contains information which will be used after you complete the LASSI.**

## Scoring Directions

After responding to statements 1-77, you may begin the scoring process. Peel off pages 2 and 3 of the inventory. These are the pages you marked with your answers. When the pages are removed, you will then see pages 4 and 5 of the inventory. These pages contain copies of the responses you made to the LASSI statements. Notice that each item is accompanied by a number you darkened and a three-letter code, such as ANX. You will use the code for each item as well as your answer to that item in calculating and plotting your scores.

To calculate your scores for the LASSI, you will need to add the numbers that have been darkened for each of the 10 different scales. Write the darkened number for each scale item in the appropriate space below.

For example, look at the first scale, labeled ATT below. The first item number for the ATT scale is item #5. Go to page 4 and find item #5. Copy the darkened number, in this example the number 3 (e.g. 1 2 ■ 4 5), into the space above item (5) on this page. Now find the next item for that scale, item #14. Write the darkened number from page 4 in the space provided.

Do this for all items for the ATT scale. Then carefully add the numbers and write the total at the far right in the space provided. **You will use these numbers again so please double check your work carefully.**

Now finish copying the darkened numbers for each item for all the scales below. Don't forget to add the numbers for each scale.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ATT | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ | = ___ | ATT |
| Item# | (5) | (14) | (18) | (29) | (38) | (45) | (51) | (69) | | |
| MOT | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ | = ___ | MOT |
| Item# | (10) | (13) | (16) | (28) | (33) | (41) | (49) | (56) | | |
| TMT | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ | = ___ | TMT |
| Item# | (3) | (22) | (36) | (42) | (48) | (58) | (66) | (74) | | |
| ANX | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ | = ___ | ANX |
| Item# | (1) | (9) | (25) | (31) | (35) | (54) | (57) | (63) | | |
| CON | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ | = ___ | CON |
| Item# | (6) | (11) | (39) | (43) | (46) | (55) | (61) | (68) | | |
| INP | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ | = ___ | INP |
| Item# | (12) | (15) | (23) | (32) | (40) | (47) | (67) | (76) | | |
| SMI | ___ + | ___ + | ___ + | ___ + | ___ | | | | = ___ | SMI |
| Item# | (2) | (8) | (60) | (72) | (77) | | | | | |
| STA | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ | = ___ | STA |
| Item# | (7) | (19) | (24) | (44) | (50) | (53) | (62) | (73) | | |
| SFT | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ | = ___ | SFT |
| Item# | (4) | (17) | (21) | (26) | (30) | (37) | (65) | (70) | | |
| TST | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ + | ___ | = ___ | TST |
| Item# | (20) | (27) | (34) | (52) | (59) | (64) | (71) | (75) | | |

**CAUTION - There should be nothing between this two-part form and your desktop.**

Very much typical of me
Fairly typical of me
Somewhat typical of me
Not very typical of me
Not at all typical of me

1. I worry that I will flunk out of school. a b c d e
2. I am able to distinguish between more important and less important information during a lecture. a b c d e
3. I find it hard to stick to a study schedule. a b c d e
4. After a class, I review my notes to help me understand the information. a b c d e
5. I don't care if I finish school as long as I find a husband/wife. a b c d e
6. I find that during lectures I think of other things and don't really listen to what is being said. a b c d e
7. I use special study helps, such as italics and headings, that are in my textbook. a b c d e
8. I try to identify the main points when I listen to lectures. a b c d e
9. I get discouraged because of low grades. a b c d e
10. I am up-to-date in my class assignments. a b c d e
11. Problems outside of school – being in love, financial difficulties, conflict with parents, etc. – cause me to neglect my school work. a b c d e
12. I try to think through a topic and decide what I am supposed to learn from it rather than just read it over when studying. a b c d e
13. Even when study materials are dull and uninteresting, I manage to keep working until I finish. a b c d e
14. I feel confused and undecided as to what my educational goals should be. a b c d e
15. I learn new words or ideas by visualizing a situation in which they occur. a b c d e
16. I come to class unprepared. a b c d e
17. When preparing for an exam, I create questions that I think might be included. a b c d e
18. I would rather not be in school. a b c d e
19. My underlining is helpful when I review text material. a b c d e

Very much typical of me
Fairly typical of me
Somewhat typical of me
Not very typical of me
Not at all typical of me

20. I do poorly on tests because I find it hard to plan my work within a short period of time. a b c d e
21. I try to identify potential test questions when reviewing my class material. a b c d e
22. I only study when there is the pressure of a test. a b c d e
23. I translate what I am studying into my own words. a b c d e
24. I compare class notes with other students to make sure my notes are complete. a b c d e
25. I am very tense when I study. a b c d e
26. I review my notes before the next class. a b c d e
27. I am unable to summarize what I have just heard in a lecture or read in a textbook. a b c d e
28. I work hard to get a good grade, even when I don't like a course. a b c d e
29. I often feel like I have little control over what happens to me in school. a b c d e
30. I stop periodically while reading and mentally go over or review what was said. a b c d e
31. Even when I am well prepared for a test, I feel very anxious. a b c d e
32. When I am studying a topic I try to make everything fit together logically. a b c d e
33. I talk myself into believing some excuse for not doing a study assignment. a b c d e
34. When I study, I have trouble figuring out just what to do to learn the material. a b c d e
35. When I begin an examination, I feel pretty confident that I will do well. a b c d e
36. When it comes to studying, procrastination is a problem for me. a b c d e
37. I check to see if I understand what the instructor is saying during the lecture. a b c d e
38. I do not care about getting a general education, I just want to get a good job. a b c d e

**CAUTION - There should be nothing between this two-part form and your desktop.**

Very much typical of me
Fairly typical of me
Somewhat typical of me
Not very typical of me
Not at all typical of me

39. I am unable to concentrate well because of restlessness or moodiness. a b c d e
40. I try to find relationships between what I am learning and what I already know. a b c d e
41. I set high standards for myself in school. a b c d e
42. I end up "cramming" for almost every test. a b c d e
43. I find it hard to pay attention during lectures. a b c d e
44. I key in on the first and/or last sentences of most paragraphs when reading my text. a b c d e
45. I only study the subjects I like. a b c d e
46. I am distracted from my studies very easily. a b c d e
47. I try to relate what I am studying to my own experiences. a b c d e
48. I make good use of daytime study hours between classes. a b c d e
49. When work is difficult I either give up or study only the easy parts. a b c d e
50. I make drawings or sketches to help me understand what I am studying. a b c d e
51. I dislike most of the work in my classes. a b c d e
52. I have trouble understanding just what a test question is asking. a b c d e
53. I make simple charts, diagrams, or tables to summarize material in my courses. a b c d e
54. Worrying about doing poorly interferes with my concentration on tests. a b c d e
55. I don't understand some course material because I don't listen carefully. a b c d e
56. I read textbooks assigned for my classes. a b c d e
57. I feel very panicky when I take an important test. a b c d e
58. When I decide to study, I set aside a specific length of time and stick to it. a b c d e
59. When I take a test, I realize I have studied the wrong material. a b c d e

Very much typical of me
Fairly typical of me
Somewhat typical of me
Not very typical of me
Not at all typical of me

60. It is hard for me to decide what is important to underline in a text. a b c d e
61. I concentrate fully when studying. a b c d e
62. I use the chapter headings as a guide to identify important points in my reading. a b c d e
63. I get so nervous and confused when taking an examination that I fail to answer questions to the best of my ability. a b c d e
64. I memorize grammatical rules, technical terms, formulas, etc., without understanding them. a b c d e
65. I test myself to be sure I know the material I have been studying. a b c d e
66. I put off studying more than I should. a b c d e
67. I try to see how what I am studying would apply to my everyday living. a b c d e
68. My mind wanders a lot when I study. a b c d e
69. In my opinion, what is taught in my courses is not worth learning. a b c d e
70. I go over homework assignments when reviewing class materials. a b c d e
71. I have difficulty adapting my studying to different types of courses. a b c d e
72. Often when studying I seem to get lost in details and "can't see the forest for the trees." a b c d e
73. When they are available, I attend group review sessions. a b c d e
74. I tend to spend so much time with friends that my coursework suffers. a b c d e
75. In taking tests, writing themes, etc. I find I have misunderstood what is wanted and lose points because of it. a b c d e
76. I try to interrelate themes in what I am studying. a b c d e
77. I have difficulty identifying the important points in my reading. a b c d e

# Plot Your Scores - *Student's Copy*

Name: ______________________
Date: ______________________
I.D.#: ______________________

The chart below is used to interpret the scores you calculated on page 2 of this booklet. Each column of the table below is labeled using the three-letter codes. Copy your scores from page 2 into the space provided for each scale. Find your score on the scale directly above each scale code and place an X over this number. Do this for each scale.

For example, if your ATT score was 29, find the number 29 on the set of numbers just above the ATT scale name and place an X over the 29, as shown in the example below.

| | |
|---|---|
| 40 | 31 |
| 35 | 30 |
| 30 | $\cancel{29}$ |
| 25 | -- |

If you cannot find your exact score, place an X over the next lowest number. When you have finished all 10 scale scores, connect the X's to see your learning and study strategies profile.

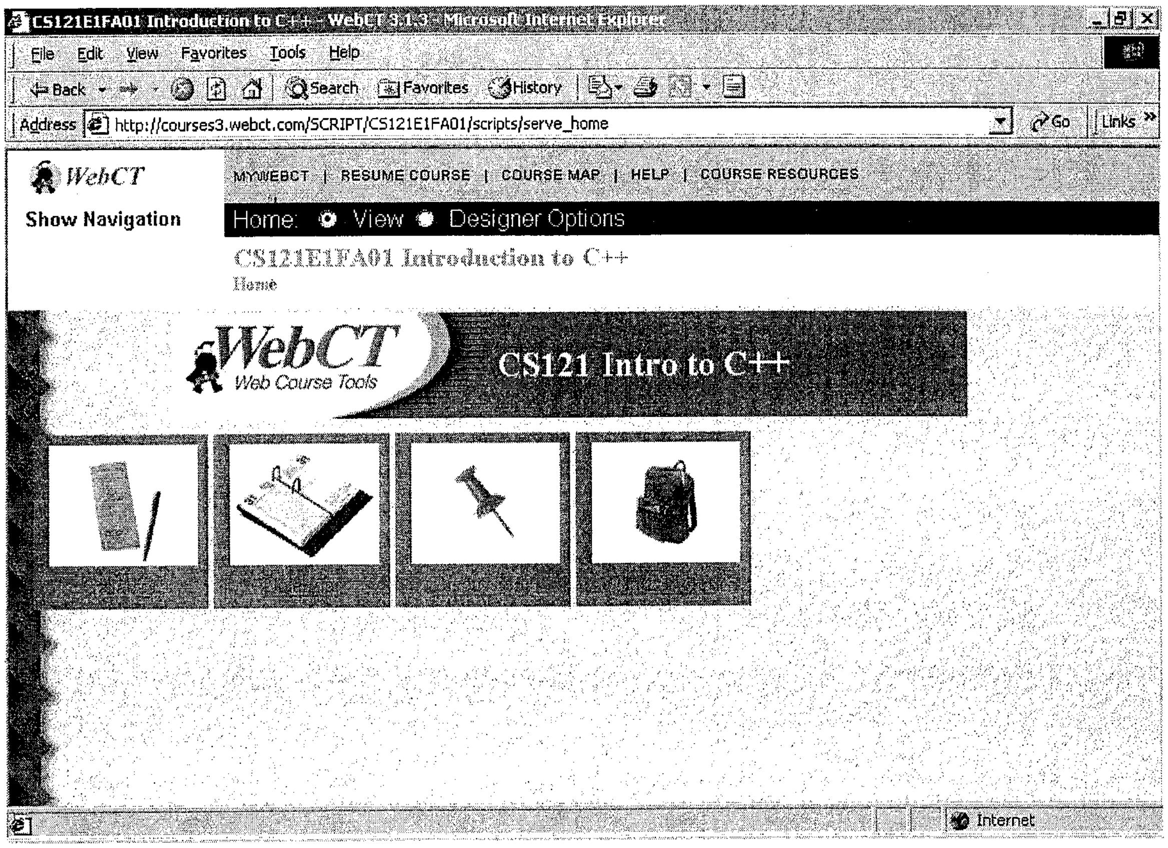
The columns on the far left and far right of the chart show percentiles. You can use these percentiles to look at your scores in relation to other college students answering the same items.

Each of the three-letter codes indicates a category of learning and study strategies or methods. The meanings of the codes are:

| | |
|---|---|
| **ATT** | • attitude and interest |
| **MOT** | • motivation, diligence, self-discipline, and willingness to work hard |
| **TMT** | • use of time management principles for academic tasks |
| **ANX** | • anxiety and worry about school performance |
| **CON** | • concentration and attention to academic tasks |
| **INP** | • information processing, acquiring knowledge, and reasoning |
| **SMI** | • selecting main ideas and recognizing important information |
| **STA** | • use of support techniques and materials |
| **SFT** | • self testing, reviewing, and preparing for classes |
| **TST** | • test strategies and preparing for tests. |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 99 | 39 | 39 | 39 | 39 | 38 | 39 | 25 | 38 | 39 | 39 | 99 |
| 95 | 38 | 38 | 33 | 36 | 34 | 36 | 23 | 33 | 33 | 37 | 95 |
| 90 | 37 | 37 | 32 | 34 | 32 | 34 | 22 | 31 | 32 | 35 | 90 |
| 85 | 36 | 36 | 30 | 33 | 31 | 32 | 21 | 30 | 30 | 34 | 85 |
| 80 | 35 | 35 | 29 | 32 | 30 | 31 | -- | 29 | 29 | 33 | 80 |
| **75** | **--** | **--** | **28** | **31** | **29** | **30** | **20** | **28** | **--** | **--** | **75** |
| 70 | 34 | 34 | 27 | 30 | -- | 29 | -- | 27 | 28 | 32 | 70 |
| 65 | -- | 33 | 26 | 29 | 28 | -- | 19 | 26 | 27 | -- | 65 |
| 60 | 33 | 32 | 25 | 28 | 27 | 28 | -- | -- | -- | 31 | 60 |
| 55 | -- | -- | 24 | 27 | 26 | 27 | -- | 25 | 26 | -- | 55 |
| **50** | **32** | **31** | **23** | **26** | **25** | **--** | **18** | **--** | **25** | **30** | **50** |
| 45 | -- | 30 | 22 | 25 | 24 | 26 | -- | 24 | -- | 29 | 45 |
| 40 | 31 | -- | 21 | 24 | 23 | 25 | 17 | 23 | 24 | -- | 40 |
| 35 | 30 | 29 | 20 | 23 | 22 | 24 | -- | -- | 23 | 28 | 35 |
| 30 | 29 | 28 | 19 | 22 | 21 | 23 | 16 | 22 | 22 | 27 | 30 |
| 25 | -- | 27 | 18 | 21 | 20 | 22 | -- | 21 | 21 | 26 | 25 |
| 20 | 28 | 26 | 17 | 20 | 19 | 21 | 15 | 20 | 20 | 25 | 20 |
| 15 | 27 | 25 | 15 | 19 | 18 | 20 | 14 | 19 | 19 | 24 | 15 |
| 10 | 25 | 23 | 14 | 17 | 16 | 19 | 13 | 18 | 18 | 22 | 10 |
| 05 | 23 | 20 | 12 | 15 | 13 | 17 | 11 | 16 | 16 | 19 | 05 |
| 01 | 19 | 17 | 09 | 12 | 10 | 14 | 08 | 13 | 12 | 14 | 01 |
| | ___ | ___ | ___ | ___ | ___ | ___ | ___ | ___ | ___ | ___ | |
| | ATT | MOT | TMT | ANX | CON | INP | SMI | STA | SFT | TST | |

## Appendix C


CS121E1FA01 Introduction to C++ - WebCT 3.1.3 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://courses3.webct.com/SCRIPT/CS121E1FA01/scripts/serve_home
Go Links
WebCT
Show Navigation
MYWEBCT | RESUME COURSE | COURSE MAP | HELP | COURSE RESOURCES
Home: View Designer Options
CS121E1FA01 Introduction to C++
Home
WebCT Web Course Tools
CS121 Intro to C++
Internet

## Syllabus

### Instructor Information

| | |
|---|---|
| Name | Karen M. Mattison |
| Email | mattison@salemcc.org |
| Office location | Tillis Hall, rear of S107 |
| Office hours | M/W 09:30am- 11:00am, or by appt. |
| Phone | (856) 299-2100 Ext. 672 |
| Biography | Please refer to Faculty Web Page that is managed by SCC Webmaster! www.s |
| Teaching assistants | none at present |

### Course Information

| | |
|---|---|
| Course title | Introduction to C++ |
| Course number | CS121 |
| Course discipline | Computing Sciences |
| Course description | This course will introduce the student to computer programmin using the C++ p algorithm design, implementation, function calls and function headings, class m definitions, selection structures, repeating structures, file streams, and vectors. |
| Course date | Jan 14, 2002 through May 6, 2002 |
| Location | TBA |
| Meeting day (s) | TBA |
| Meeting time(s) | TBA |
| Prerequisite (s) | College level placement or the instructor's approval. |

### Course Goals

Course Goals

```
I.   Problem Analysis and Algorithm
     Design    Week Two

II.  Implementation Issues    Week Four

III.Functions    Week Six

IV.  Classes and Objects    Week Eight

V.   Selection Structures    Week Ten
```

VI. Repetition Structures Week Twelve

VII.File Streams Week Fourteen

VIII.Vectors Week Sixteen

## Policies

| | |
|---|---|
| Introduction | Please refer to the following SCC link for all formal policies and procedures. ww |
| Additional information | Purchase all your SCC books online at: shop.efollett.com/htmlroot/storehome/s the discussion board 3X per week. All students will be required to refer to the c |

## Textbooks

| | |
|---|---|
| Required reading | *Computing Fundamentals with C++*, Rick Mercer, Franklin, Beedle & Associate |

## Ch01 - Analysis and Design

Lesson: Lesson One

Objectives or Goals:

To understand what program development is.

To understand the characteristics of a good algorithm, its pattern, an

To understand that a deliverable will be expected from each phase of t

To understand the relationship between a object and the class it is ap

To understand that an object will have a name, state, and a set of ope

Topics:

```
Program Development phases
      - Analysis
      - Design
      - Implementation
```

Readings: Page 1 - Page 19 from the textbook

Assignments: Complete the Self-Check on page 19.

In addition, the student is to complete project 1B on page 26.

## Ch02 - Implementation

Lesson: Lesson One

Objectives or Goals:

- Understand how to include existing source code in your program
- Obtain data from the user and display information to the user
- Evaluate and create arithmetic expressions
- Understand that these common operations are available to many object

| | |
|---|---|
| Topics | -The C++ source program (.cpp)<br>  -Its standard format<br>  -Tokens: the smallest pieces of a pgm.<br>  -Categories of tokens: special symbols, keywords, identifiers, & constants.<br>-Commenting your source code<br>-Display information to monitor (cout)<br>-Obtain information from keyboard (cin)<br>-Basic arithmetic using assignment (=) & arithmetic operators (+, -, *, /). |
| Readings | Ch01 p20 - p23 & Ch02 p30 - p52 |
| Assignments | p 27, 1F<br>p 63, 2H |

## Ch03 - Function Calls and Headings

| | |
|---|---|
| Lesson | Lesson One |
| Objectives or Goals | - Evaluate a few mathematical and trigonometric functions<br>- Use arguments in function calls<br>- Appreciate why programmers divide software into functions<br>- read function headings so you can use existing functions<br>- use integer objects<br>- use quotient remainder integer division<br>- understand the categories of errors that occur during the implementa |
| Topics | cmath Functions<br>using standard library functions in source code<br>documentation for a function: Preconditions & Postconditions<br>Implementation Errors & Warnings |
| Readings | p 65 - 96 |
| Assignments | p 106, 3H   U.S. Change |

## Ch04 - Messages and Member Functions

| | |
|---|---|
| Lesson | Lesson One |
| Objectives | -send messages to objects |

| | |
|---|---|
| or Goals | -send a few string and ostream messages and understand their effects<br>-problem solve with grid and bandAccount objects<br>-apprieciate why programmers partition software into classes, which ar |
| Topics | -Modeling the Real World with UML<br>-Class and Object diagrams<br>-Standard Messages and Member Functions<br>-Class Member Functions |
| Readings | pp 108 - 135 |
| Assignments | 4D Two Bank Accounts |

**Ch05 - Functions and Parameters**

| | |
|---|---|
| Lesson | Lesson One |
| Objectives or Goals | - implement nonmember function - pass values to functions as input - return val |
| Topics | Implementation of Non-member functions The Return statement Test Drivers A |
| Readings | pp 145-186 |
| Assignments | 5M |

**Ch06 - Class Definitions & Member Functions**

| | |
|---|---|
| Lesson | Lesson One |
| Objectives or Goals | - read & understand class definitions - implement class member functions - obje |
| Topics | Class Definitions public vs. private Class member functions Accessor vs. Mutat |
| Readings | pp 191 - 232 |
| Assignments | pp 208/209 6A, 6B, 6C pp 230 6F |

**Ch07 - Selection**

| | |
|---|---|
| Lesson | Lesson One |
| Objectives or Goals | + recognize when to use the Guarded Action pattern (do something only under<br>C++ if statement + use relational operators such as >, >=, <, <=, ==, != + create<br>"or" using these symbols !, &&, \|\| + use bool objects + understand the Alternativ<br>C++ + implement the Multiple Selection pattern with if...else and switch + solve |
| Topics | - Selective Control - Logical Expressions with Relational operators - The Altern<br>Objects - A bool Member Function - Multiple Selection - The switch Statement |
| Readings | pp 234 - 281 |
| Assignments | p 267 assignment 7H p 290 assignment 7M |

**Ch08 - Repetition**

| | |
|---|---|
| Lesson | Lesson One |
| [illegible] | [illegible] |

| | |
|---|---|
| Objectives or Goals | Recognize and use the Determinate Loop pattern to execute a set of statement the C++ for statement Recognize and use the Indeterminate Loop pattern to ex indeterminate loops with the C++ while statement. |
| Topics | Repetitive Control Application of the Determinate Loop Pattern Algorithmic Patt statement The for statement |
| Readings | pp 293- 339 |
| Assignments | Please refer to course calendar for you final assignment that is due on the last |

## Compiled Calendar Entries

Compile | Calendar

### January 2002

| Date | Event |
|---|---|
| January 14 | First Day of Classes |
| January 18 | First Discussion Details<br>- The first discussion question is posted and must be responded to by each student registered in this course. If no posting is seen by this date the student will receive a zero (0) grade for this discussion. |
| January 21 | SCC Closed<br>- Martin Luther King\, Jr. Day - College Closed |
| January 25 | Programming Assignment One from Ch01<br>- Assignment 1F is due today! - Chapter 01\, Assignment 1F |

### February 2002

| Date | Event |
|---|---|
| February 1 | Discusson for Lesson Two is now due!<br>- Your participation in Lesson Two discussion is now expected. |
| February 8 | Programming Assignment for Chapter Two is now due!<br>- pp 63\, 2H "Wholesale Cost" |
| February 11 | Test One is now due!<br>- Test on material up to and including Ch02 is due! |
| February 15 | Lesson Three (Chapter 03) Discussion due.<br>- Discussion posting is now required for Lesson 03 (CH03). |
| February 18 | SCC Closed<br>- Presidents' Day Holiday - College Closed |
| February 22 | CH03 Programming Assignment is due<br>- p 106\, 3H U.S. Change Must have source (free of errors)\, test-script of program execution\, Analysis & Design deliverables done on MS Word. |

### March 2002

| Date | Event |
|---|---|
| March 1 | Chapter five Discussion posting due<br>- Please complete Chapter Five Discussion Posting today! |
| March 8 | Chapter 05 Assignment is do today!<br>- p.170\, 5G Payment & p 189\, 5L Display bankAccounts |
| March 12 | SCC Closed<br>- In-Service Day - No Classes |

| | |
|---|---|
| March 13 | SCC Closed<br>- Spring Break - No Classes |
| March 14 | SCC Closed<br>- Spring Break - No Classes |
| March 15 | SCC Closed<br>- Spring Break - No Classes |
| March 20 | Last Day to Withdraw from Classes Without Academic Penalty |
| March 22 | Test II must be taken by today !<br>- Test II is now available and covers text materail up to and including Ch04. |
| March 27 | CH06 Discussion Posting is due today! |
| March 28 | Last Day to File Petition for Graduation |
| March 29 | Spring Holiday - College Closed |
| March 30 | Spring Holiday - College Closed |
| March 31 | Spring Holiday - College Closed |

## April 2002

| **Date** | **Event** |
|---|---|
| April 1 | Spring Holiday - College Closed |
| April 5 | CH06, Part A Assignment is due today!<br>- PP 207 & 208 Programming projects 6A\, 6B\, 6C |
| April 12 | CH06, Part B assignment is due today!<br>- PP 230 & 231 The Counter Class |
| April 13 | TEST III should be taken by Mon. at 12 midnight! (CH05 & CH06) |
| April 19 | CH07 discussion posting is due today! |
| April 26 | CH07, Part A assignment is due today!<br>- P 267\, 7H Safe Accounts |

## May 2002

| **Date** | **Event** |
|---|---|
| May 3 | CH07, Part B assignment is due today!<br>- P 290\, 7M Student |
| May 4 | TEST IV will need to be taken by Mon. evening at 12 midnight! (CH7 & CH8) |
| May 6 | Last Day of Classes<br>Final programming project is due today!<br>- Write a final programming project that will call 4 free functions. The name of each function is up to you. However\, the first must display the powers of 2 (Binary). The second\, the powers of 8 (Octal). The third\, the powers of 16 (Hexadecimal) and the last the powers of 10 (Decimal). The end-user should be able to provide a number and based on this number\, each of the above mentioned functions should be called that number of |

times. An example of the output follows\, assuming the number 5 was entered. Count Binary Octal Hexadecimal Decimal 1 2 8 16 10 2 4 64 256 100 3 8 512 4096 1000 4 16 4086 65536 10000 5 32 32768 1048576 100000

---

Compile Calendar

# Visual C++ IDE Reference

## Overview

The Visual C++ IDE (Integrated Development Environment) allows you to write many different kinds of software, from simple console programs to DLLs (Dynamic Link Libraries) and Active Template Library Components. (No need to worry if you don't even know what these are.) A console program is one that has a simple (old-fashioned, if you like) text-based interface, in which program output appears on lines that scroll upward to the top of, and eventually off, the screen.
Whenever you do something in Visual C++, you generally do it within Visual C++'s notion of *workspaces* and *projects*. You can have several workspaces and several projects within each workspace, but for simplicity we will generally have one project per workspace. Visual C++ does its best to help you out when you're working. Sometimes this can be pleasant and useful. Other times it can be downright unpleasant and frustrating. It can also be quite overwhelming, when you find, for example, that Visual C++ has created a dozen or so files in all, just in the process of helping you create a very simple program (such as a typical "Hello, world!" program, for example).
So, it is important to have a good sense of the big picture, and not to be overwhelmed or intimidated by all the minutiae that need not concern you, at least for the forseeable future.

## Getting Started: Creating, Building, and Running a New "Win32 Console Application"

1. Start Visual C++ from the appropriate submenu of the Start menu.
   You will probably get an interface containing a menu bar, one or more "toolbars", and two (or possibly three) blank subwindows.

2. In any case, select File | New (or press Ctrl+N).
   A "New" window with the "Projects" tab pre-selected should appear, and you should be presented with the opportunity to choose what kind of project you want.

3. Select "Win32 Console Application" from the "Projects" tab.

4. Enter a pathname for your project in the "Location:" box.
   The (apparently unchangeable) default entry for this box is a part of the C: drive to which you cannot write. So, change it to a location on your A: drive. A simple solution is just to enter A:\, but you may want to think more about where on your A: drive you want the project to be located.

5. Enter a name for your project in the "Project name:" box, and note how it is automatically added to the end of your pathname.
   This is generally useful but can sometimes be confusing and cause you to end up with a path to your project containing two subdirectories with the same name. Just watch what you're doing.

6. Make sure that the "Create new workspace" radio button is activated.

7. Make sure that "Win 32" is checked in the "Platforms:" box.

8. Click OK.

9. In the next window make sure that "An empty project" is chosen.

10. Click "Finish", and read what's in the "New Project Information" window.

11. Click OK.
    The "ClassView" pane in the workspace window at the left (or upper left) should become active at this point. You can switch to the FileView pane in this window if you like, and you will want to later, but since you don't yet have any classes or files there's not much point.

At this point you have a single workspace and a single project within that workspace, and you are ready to begin adding files to this project. If you have an already existing file that you wish to add to the project, choose "Add to project | Files..." menu item from the "Project" menu. In the dialog box that opens make sure the "Files of type:" box shows the kind of file you're looking for. Then browse to that file and click OK. Alternatively, enter the full pathname to the the file in the "File name:" box and then click OK. Now you can check in the FileView pane of the workspace window to make sure the file is listed there and the project is thus aware of it.
The steps which follow, however, take a different approach, though they continue on from the numbered steps above. They assume you are going to type in a source code file, as you would now do if you were creating a first, single-file program, such as a typical "Hello, world!" program.

12. Again, as you did in step 2 above, select File | New or press Ctrl+N.
    This time the "New" window that appears should have the "Files" tab pre- selected.

13. Select the type of file you wish to create by clicking on it.
    For console programs this will normally be either "C++ Source File" or "C/C++ Header File".

14. Enter a name for the file in the "File name:" box.
    This file will be saved in your project folder, and default extensions (.cpp or .h as appropriate) will be supplied if you do not give one yourself.

15. Click OK.
    The text editing window becomes active and ready for text entry.

16. Type in the contents of the file.

17. Repeat steps 12-16 for each file that you wish to add to the project in this way.
    Note that files added in the other way discussed above can be combined with files added in this way as part of the same project.

18. When all files are ready, select "Build ProjectName.exe" on the "Build" menu, or press F7, or click the build icon on the (unlabeled) "Build mini-bar".
    Note that the menu option contains the actual name of your project, followed by a .exe extension (and not the name of any particular file, including the file with the main function in it, unless this name happens to coincide with the name of your project). The "Output" window at the bottom of the screen will open (if it is not already open) and any compiler or linker errors you may have will be reported in this window. Correct any errors and rebuild if necessary. Double clicking on the line containing the error in the output window will place a pointer at (with luck) or near (more likely) the error in your source code. For more details on any particular error, select the error number and press F1.

19. When the project "builds" successfully (i.e., when it compiles and links without any errors), you may then "run" the resulting executable by choosing "Execute ProjectName.exe" from the "Build" menu, or press Ctrl+F5, or click the red exclamation mark (!) on the "Build mini-bar". A DOS window will then open, and your program will run. When it finishes running, a message to "Press any key to continue" will appear, and pressing any key will cause that window to disappear as you are returned to the Visual C++ environment.

20. You can, if you wish, now modify your program, then re-build and re-run. Or, if you make no further changes, you may now close Visual C++, and it will shut down immediately, since all of your files will have been automatically saved as a part of the build process. After you do shut down, open Windows Explorer and examine the list of files created in your project directory and in a subdirectory of your project directory called "Debug" (in which your executable lives). For the most part you can ignore the many additional files created by Visual C++ as it builds your project. If you want further details, see below under **Files Used/Created by Visual C++**.

Once you have an executable produced with Visual C++ you can run the program independently of Visual C++ in the following two ways:

- You can double-click on the executable file while in Windows Explorer. When you do this, note that if the program does not read from the keyboard you may "miss" its running since the DOS window in which it runs may open and close very quickly. The reason for this is that Windows Explorer does not supply that "Press any key to continue" facility that you saw in the Visual C++ IDE.
- You may also open a DOS window yourself (by choosing Start | Run, entering `cmd` and pressing Return) and then run the program directly from the command line, in the "usual" way.

## Working with Files in a Project

Once you have added to your project all the files the project needs, you may perform any of the following actions.

1. Double click on the name of a source code file in the file view of the workspace window to display it in the text editor window.

2. Place the cursor in the word "include" (of any #include), right click, and then choose "Open Document ..." to view that particular header file in the text editor window.

3. Press F7, click the corresponding menu item on the "Build" menu, or click the Build icon (the one with the two downward-pointing blue arrows) on the Build mini-toolbar to "build" the executable for the project. The IDE is smart enough to build only what needs to be re-built (because it has changed) if this is not the first build and your project consists of several "pieces". Note that files are saved to disk as part of the build process. If you get build errors, they will show up in the Output window at the bottom, and you can double click on an error to show where it has occurred in your source code or highlight the error and press F1 to get more information on the error from the Help system.

4. Choose "Rebuild All" from the Build menu if you wish to re-build everything in any case (whether it has changed or not).

5. Press Ctrl+F7 to just compile the file named in this option of the Build menu (which will be either the file currently active in the text editor, or the one whose name you've selected in the

workspace window if you have in fact selected one there). You may also click the leftmost icon (the one with the single arrow pointing downward) on the Build mini-toolbar to do this.

6. Choose the "Execute" option (marked with a red exclamation mark) from the Build menu or click on the red exclamation mark on the Build mini-toolbar to run your program, once it has built successfully. A potentially confusing situation here is this: If your program reads input from a file, the IDE expects that file to be in your project directory, even though the executable is in a subdirectory called Debug in your project directory. Such a data file does not have to be added to the project, so long as it is present and in the right location.

7. Click on any file in the workspace window file view and press Delete to delete that file from the project (though *not* from its location on disk).

## The Text Editor: Creating and Modifying Source Code Files

The first thing to know about the Visual C++ text editor is that many of the "usual" commands for navigation and applying changes to text that are familiar to users of other Microsoft products such as Word also work here in the same way. Hence, assuming this familiarity, we shall not give the details of those commands here. But it is assumed that you are familiar with what the following keyboard commands do for you when editing: the four arrow keys, Home, End, Ctrl+Home, Ctrl+End, PageUp, PageDown (and using the Shift key with any of these motion commands to select text), and also Ctrl+C, Ctrl+X, Ctrl+V, Ctrl+A, and Ins.

We list below some commands and features that are either not present in other applications, or are not as widely known or used, but which you may find useful in the Visual C++ environment:

1. 1. 1. Let the editor's "smart autoindent" feature work for you. It knows, for the most part, how your code should be formatted, and you should let it decide, for example, things like what the indent level should be (4 spaces), and the proper placement of braces. To make sure this feature is "on", choose Tools | Options ... | Tabs and activate the "Smart" option of the "Auto indent" group if this is not the case already. In fact, you can select a section of code that has had its formatting "messed up" or is not formatted quite right and have the whole thing formatted for you automatically by pressing Alt+F8. This action depends on the "context", however, and assumes that the surrounding code has been formatted properly, since it bases the new formatting on this surrounding code.

2. 2. 2. The F8 key has other users: F8 by itself toggles a character select, Ctrl+F8 toggles a line select, and Ctrl+Shift+F8 toggles a block select.

3. 3. 3. If a group of contiguous lines of code need to be indented or unindented one or more levels, select those lines of code and then press the TAB key to indent an additional level, or the Shift+TAB combination to unindent one level.

4. 1. 4. It is a good idea to have *no* TAB characters in your file. You can remove all TAB characters from a file at any time by selecting your entire file with Ctrl+A and then choosing Edit | Advanced | Untabify Selection. Even better, before beginning text entry, choose Tools | Options and then on the Tabs pane make sure the "Insert spaces" radio button is activated, *not* the "Keep tabs" button. And by the way, this is also where you set (or reset, if necessary) the "Tab size" and "Indent size", both of which should be 4.

5. 2. 5. Ctrl+Shift+8 toggles the display of whitespace in your source code. Blank spaces show up as a dot, while each Tab character shows up as two greater than signs (>>).

6. 3. 6. Double-clicking on the little rectangular box above the up-arrow at the top of the scroll-box region at the right of the text editor window will split the current window into two equal parts horizontally. This gives you two views of the same file, which can be scrolled and edited independently. Double-clicking on the separating bar will return you to a single window. Choosing the Split option from the Window menu divides the current window into four, which is probably overkill. Double-clicking on the intersection at the center returns you to a single window.

7. 4. 7. Ctrl+F2 toggles an unnamed bookmark at the cursor, and F2/Shift+F2 move to the next/previous bookmark. To use named bookmarks, choose Edit | Bookmarks ... and follow instructions. Ctrl+Shift+F2 clears all bookmarks in the current window.

8. 5. 8. Ctrl+G allows you to go to various locations, including (probably most usefully) a particular line number.

9. 6. 9. When the cursor is on either side of a bracket (round, square, curly, or angle), Ctrl+] will place the cursor on the matching symbol.

10. 7. 10. 10.Ctrl+U/Ctrl+Shift+U converts selected text to lower/upper case.

11. 8. 11. 11.To change the font size, choose Tools | Options ... | Format | Size:.

12. 9. 12. To expand a source file window to full screen, choose View | Full Screen. Press Esc to return to the current window at the previous size. You can use other commands and/or menus while in full screen mode if you know their keyboard equivalents.

13. 10. 13. Ctrl+Z undoes the previous change. Pressing it again undoes the change before that, and so on. If you undo one change to many, you can "re-do" the last undo with a Ctrl+Y.

14. 11. 14. Clicking in the grey area to the left of the edit window selects the line opposite. Ctrl+Click there selects the entire contents of the current window (visible or not). In other words, this action has the same effect as Ctrl+A.

15. 12. 15. Tab/Shift+Tab indent/unindent selected text one level of indentation.

16. 13. 16. Ctrl+Tab cycles through the queue of open windows, much like Alt+Tab does in Windows itself.

17. 14. 17. To choose what additional information you want on the page when you print a file, as well as the margins you want, go to File | Page Setup ...

18. 15. 18. Ctrl+F opens a Find dialog box for entry of a search string, after which F3/Shift+F3 find the next/previous instances. Esc gets you out of Find and back to the source code window. Also, Ctrl+F3 will start a search for selected text, and Ctrl+H starts a Find/Replace.

19. 16. 19. Ctrl+I starts and "incremental search", i.e., a search that starts as soon as you begin typing and looks for the first instance of as much as you have typed in so far, and Ctrl+Shift+I starts a similar backwards search.

20. 17. 20. F4/Shift+F4 move to the next/previous build error or find-in-files match.

21. 18. 21. Ctrl+Shift+T interchanges the current and previous words, while Alt+Shift+T interchanges the current and previous lines.

22. 19. 22. Ctrl+Delete deletes to the beginning of the next word, Ctrl+Backspace to the beginning of the previous word.

23. 20. 23. Ctrl+L deletes the current line and places it on the clipboard, Ctrl+Shift+L deletes it without placing on the clipboard (i.e., its gone). Alt+Shift+L deletes to the end of the line.

24. 21. 24. Ctrl with the arrow keys provides some useful commands: Ctrl+RightArrow moves the cursor to the beginning of the next word, while Ctrl+LeftArrow moves the cursor to the beginning of the previous word. Ctrl+UpArrow scrolls the window down one line, while Ctrl+DownArrow scrolls the window up one line.

```
//Programmer Name:  Karen Mattison
//Date:             00/00/00
//Brief Desc:       This is your first C++ program.  It will be used to
//                  demo how to create a project in C++, add one file,
//                  and execute.



#include


using namespace std;


int main( )
{


    cout <<    "\n\t H E L L O    C L A S S \n " ;

      cout <<  "\n\t Welcome to C++ \n"   ;



      return 0;

}
```

The 3 step process in program development will include analysis, design, and implementation. The following identifies which tasks are performed during each phase.

| **Phase** | **Activities** |
|---|---|
| Initiaion | Specify the problem |
| Analysis | Read/ Understand the problem |
| Design | Look for patterns to guide algorithm development |
| Implementation | Translate the design into a programming language.<br>Fix errors.<br>Test the program. |
| Maintenance | Update the program or enhance the program. |

1E Weighted Average

Analysis Phase

| Problem | Data Name Objects | Input/Output | Sample Problem |
|---|---|---|---|
| Compute a weighted average | QuizAverage | Input | 70 |
| | Midterm | Input | 80 |
| | LabGrade | Input | 90 |
| | FinalExam | Input | 100 |
| | CourseGrade | Output | 86.5 |

Design Phase

1. Write an algorithm that outlines a solution to the GPA problem.
2. Obtain QuizAverage, Midterm, LabGrade, and FinalExam
3. Compute CourseGrade = .2*QuizAverage+.3*Midterm+.35*LabGrade+.25*FinalExam
4. Display CourseGrade

```
// Programmer:              Karen Mattison
// Current Date:            1-30-02
// Brief Description:       Chapter 02 Program Assignment
//                                       Deli example


#include

using namespace std;

int main()
{

      int choice;
      double price;

      // Prompt

      cout << "------------------------------------------" << endl;
      cout << "     1 - Drink        $ 1.00              " << endl;
      cout << "     2 - Burger       $ 2.00              " << endl;
      cout << "     3 - Pie          $ 1.50              " << endl;
      cout << "     Make a Choice                        " << endl;
      cout << "------------------------------------------" << endl;

      // Input

      cin >> choice;

      // Processing

      if (choice == 1)
            price = 1.00;

      if (choice == 2)
            price = 2.00;

      if (choice == 3)
            price = 1.50;

      // Output

      cout << endl << "Your final bill is : " << price << endl;

      return 0;
}
```

```
//Programmer Name:   Karen Mattison
//Date:              00/00/00
//Brief Desc:        This program will execute, prompt for the number
//                   of hourse worked, the rate of pay, and then will
//                   calculate the net pay for one employee.


#include


using namespace std;


int main( )
{
//  List Objects
    double hours_worked,  rate_of_pay, net_pay;

//  Input state of Objects
      cin >> hours_worked;
      cin >> rate_of_pay;


//  Process Objects
      net_pay = hours_worked * rate_of_pay;

//  Output state of the Objects
      cout << net_pay;



      return 0;

}
```

```
//Programmer Name:   Karen Mattison
//Date:              00/00/00
//Brief Desc:        description of program


// place those header files necessary to compile here
#include
#include
#include
#include "compfun"


using namespace std;

// functions will follow.  main function is not optional.  Only 1 per
project.
int main( )
{



      return 0;

}
```

# Chapter 2: Implementation

Prerequisites: Chapter 1

Goals:

1. Teach the nuts and bolts of C++ programs: variable declaration, initialization, assignment, Input/Output, and arithmetic expressions
2. Introduce the Prompt then Input pattern
3. Let students practice implementing simple IPO algorithmic patterns
4. Evaluate and write arithmetic expressions

Experiences:

This is the chapter that focuses on the details and syntax to complete simple programs. Students are shown that a program is a collection of tokens. They write arithmetic expressions. Combined with cin and cout, students should be able to write simple Input/Process/ Output programs. With a two hour closed lab, students can complete 2 to 4 programming projects depending on experience and the choice of project. The most difficult project is 2I: Combinations, perhaps because it doesn't look familiar to most. I usually only assign it when during an engineering service course.

```
// 2F Simple Average
#include
#include
using namespace std;

int main()
{
  double t1, t2, t3;

  cout << "Enter three tests: ";
  cin >> t1 >> t2 >> t3;

  double average = (t1 + t2 + t3) / 3.0;
  cout << "Average = " << average;
}

/*
Enter three tests: 70.0 80.0 90.0
Average = 80
*/
```

# Chapter 3: Function Calls and Headings

Prerequisites: Chapters 1 and 2

Goals:

1. Get students to use existing functions, whether they are global
   functions such as sqrt
   from cmath, decimals(cout,2) from compfun, or
   bankAccount::withdraw from
   baccount.h.
2. Get students comfortable reading function heading,
   preconditions, postconditions,
3. Do int quotient/remainder division with / %

Experiences:

Writing functions and understanding argument/parameter associations is one of the more difficult concepts students encounter—besides loops perhaps—in this first course. Students have trouble reading function headings, so future programming projects are written to help ease students to understanding a program as a collection of classes that are a collection of functions. Additional support came by presenting Chapter 6: "Class definitions and Member Functions" and assigning some projects that require students to implement member functions given the class definition. Many projects give the function headings and ask students to implement the function. Students become adept at implementing free functions and member function classes. And if you get to Chapters 12 and 13 on object-oriented analysis and design, students also experience designing classes instead of implementing them.

I believe using the terms argument (in the call) and parameter (in the function heading) serves our students better than the confusing terms actual parameters and formal parameters.

I delayed coverage of programming errors to the end of this chapter to allow students to actually
experience them as they implement a few programs. This was intentional because earlier coverage had proved ineffective.

If you have computer projection, students appreciate demonstrations done with a source level
debugger that shows line by line execution of statements. You can also show the types of errors and students can commiserate.

```
// 3H: U.S Change
// Compute the minumum number of coins needed to make change
#include
using namespace std;

int main()
{
  int change;
  int half, quarter, dime, nickel, penny;

  // Input
  cout <<"Enter change [0..99]: ";
  cin >> change;

  // Process
  half    = change / 50;
  quarter = change % 50 / 25;
  dime    = change % 25 / 10;
  nickel  = change % 25 % 10 / 5;
  penny   = change % 5;

  // Output
  cout << "Half(ves)  : " << half << endl
            << "Quarter(s): " << quarter  << endl
            << "Dime(s)   : " << dime << endl
            << "Nickel(s) : " << nickel << endl
            << "Penny(ies): " << penny << endl;

  return 0;
}

/*
Enter change [0..99]: 83
Half(ves) : 1
Quarter(s): 1
Dime(s)   : 0
Nickel(s) : 1
Penny(ies): 3
*/
```

```
// Programmer Name:       K.M.Mattison
// Current Date:          2/02/02
// Brief Description:     Ch 03 ex. calculate unit_cost of pizza

#include
#include

using namespace std;

int main()
{
      double size;
      double cost;
      double radius;
      double area;
      double unit_cost;

      //double size, cost, radius, area, unit_cost;


      // Input tasks
      cout << "Enter the cost: ";
      cin >> cost;

      cout << "Enter the size: ";
      cin >> size;

      // Processing tasks

      radius = size / 2;
      area = 3.14 * pow(radius, 2);
      unit_cost =  cost / area;

      // Output tasks

      cout << "The price per square inch is $ " << unit_cost;
      cout << endl;

      return 0;
}
```

```
// Programmer Name:      K.M.Mattison
// Current Date:         2/02/02
// Brief Description:    Brief example of functions

#include

using namespace std;


//return_data_type   function_Name(pass_data_type Obj1, pass_data_type
Obj2);

int   function_One(int Age);
double function_Two(int hold_val1, double hold_val2);
void function_Three(void);


int main()
{


      function_One(25);
      function_Two(100, 1.2345);
      function_Three();
      return 0;
}

int function_One(int Age)
{
// Precondition:   assumptions about arguments
// Postcondition:  what will function do if preconditions is met

      cout << "\n I am inside function_One ";
      return 0;
}

double function_Two(int hold_val1, double hold_val2)
{
// Precondition:   assumptions about arguments
// Postcondition:  what will function do if preconditions is met

      cout << "\n The value of the first argument is: " << hold_val1;
      cout << "\n The value of the second argument is: " << hold_val2;
      return 0.0;
}

void function_Three(void)
{
// Precondition:   assumptions about arguments
// Postcondition:  what will function do if preconditions is met
      cout << "\n Function Three has no arguments to display";
}
```

# Chapter 4: Messages and Member Functions

I usually present this chapter in three 50-minute lectures using the presentations available with this instructor's manual.

This Chapter makes references to the programmer-defined classes grid and bankAccount. To
complete some of the programming projects at the end of Chapter 4, students will need the files
from their disk that accompanies the textbook or at this textbook's Web site.

http://www.cs.arizona.edu/people/mercer/compfun2/#Files

Prerequisites: Chapters 1, 2, and 3

Goals:

1. Understand the relationship between objects and classes.
2. Send messages
3. Present an appreciation of why software is divvied up into classes and functions.

The following two author-supplied classes introduced in this chapter will be used later on to illustrate new concepts:

1. bankAccount
2. grid
3. string

Because of this, I recommend that you do not skip over these classes.

My students have an easy time with bankAccount objects. They can relate to bank accounts. Bank
accounts make sense. They are not complicated. I got the idea from John Pugh in 1989 at an
ACM/SIGCSE post-symposium workshop. BankAccount has become a canonical first example,
please don't feel ashamed (as some suggest) to use it just because it is often used as a first example. Your students will not feel slighted, even if they are working adults (at least that has been my experience teaching to freshman and to many adult section in continuing education classes). Students have little problem using these bankAccount objects.

You may skip the grid class completely. Because the grid class is almost self-explanatory, students could understand examples in upcoming chapters. However, I believe it is worth the ten minutes or so to present the grid class and then to assign a few programming projects related to grid (4F, 4G, 4H, 4I, or 4J). I usually ask them to complete any two from this set of five projects. You can even ask questions on tests if you supply the class definition or diagram, or if it is open book. I don't recommend making students memorize grid messages.

The bankAccount will help when you teach students to implement their own classes. It doesn't have to be, but I use the bankAccount and grid classes to reinforce new topics later in the text. This is to avoid covering a new class for each new major topic. Future references will be made to class bankAccount and grid.

The string class is fairly easy too. Students do not seem to have little trouble with 0 being the index of character #1 in a string. Some do get confused because length returns the number of characters when they think it should return the number of characters - 1. String looks and acts like other primitive types such as int and double, except now you can send messages to string objects such as length.

The zero-based array indexing of string and grid cause little if any problem to students at this point. The big pay off comes when you present singly and doubly subscripted arrays—the vector (Chapter 10) and matrix classes (Chapter 19).

```
// 4D Two Bank Accounts
#include   // for cout
using namespace std;
#include "baccount"  // for class bankAccount

int main()
{
  bankAccount one("Melissa", 500.00);  // Construct a bankAccount object
  bankAccount theOther("Miquel", 500.00);

  one.deposit(123.45);
  theOther.deposit(50.00);

  one.withdraw(20.00);
  theOther.withdraw(60.00);

  cout << one.name() << ": " << one.balance()  << endl;
  cout << theOther.name() << ": " << theOther.balance() << endl;
  double combined = one.balance() + theOther.balance();
  cout << "Combined : " << combined << endl;

  return 0;
}

/*
Melissa: 603.45
Miquel: 490
Combined : 1093.45
*/
```

```
// 4F Pickup
#include "grid"   // for class grid

int main()
{
  grid aGrid(8, 16, 1, 7, west);
  aGrid.putDown(1, 2);
  aGrid.putDown(3, 2);
  aGrid.putDown(6, 3);
  aGrid.putDown(6, 10);
  aGrid.display();

  aGrid.move(5);
  aGrid.pickUp();
  aGrid.turnLeft();
  aGrid.move(2);
  aGrid.pickUp();
  aGrid.move(3);
  aGrid.turnLeft();
  aGrid.move(1);
  aGrid.pickUp();
  aGrid.move(7);
  aGrid.pickUp();

  cout << endl;
  aGrid.display();
  return 0;
}

/*
The grid:
. . . . . . . . . . . . . . . .
. . O . . . . < . . . . . . . .
. . . . . . . . . . . . . . . .
. . O . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
. . . O . . . . . . O . . . . .
. . . . . . . . . . . . . . . .

The grid:
. . . . . . . . . . . . . . . .
. .             . . . . . . . .
. .   . . . . . . . . . . . . .
. .   . . . . . . . . . . . . .
. .   . . . . . . . . . . . . .
. .   . . . . . . . . . . . . .
. .                 > . . . . .
. . . . . . . . . . . . . . . .
```

```
*/
```

```
// Programmer Name:       K.M.Mattison
// Current Date:          2/27/02
// Brief Description:     bankAccount Class Member Function Ex.

#include
#include

using namespace std;


class bankAccount
{

public:
      bankAccount(string Name, double Balance);
      int withdraw( double amount );
      int deposit( double amount);
      double balance(void);
      string name(void);

private:

      string my_name;
      double my_balance;

};

bankAccount::bankAccount(string Name, double Balance)
{
      // Pre-condition:
      // Post-condition:
my_name = Name;
my_balance = Balance;
}
int bankAccount::withdraw( double amount )
{
      // Pre-condition:
      // Post-condition:
        my_balance = my_balance - amount;
      return 0;
}
int bankAccount::deposit( double amount)
{
      // Pre-condition:
      // Post-condition:
        my_balance = my_balance + amount;
      return 0;
}
double bankAccount::balance(void)
{
      // Pre-condition:
      // Post-condition:
```

```
      return my_balance;
}
string bankAccount::name(void)
{
      // Pre-condition:
      // Post-condition:
      return my_name;
}

int main()
{

      bankAccount  KMM("Karen Mattison", 1000.00);

      bankAccount  SCC("SalemCommCollege", 1234.56);


      KMM.deposit( 500.00 );

      SCC.withdraw( 1000.00);

      cout << "Your account name: " << KMM.name() << endl;
      cout << "Your account balance: " << KMM.balance() << endl;

        cout << "Your account name: " << SCC.name() << endl;
        cout << "Your account balance: " << SCC.balance() << endl;

      return 0;

}
```

```
#include
#include
using namespace std;


// syntax for student class

class student {
public:
        student(string LN, string FN, int A, double GPA, string P, string
E)
      void Join_Club( string club_joined);
      void Register_for_class(string class_registered_for);
      void Pay_Bill(double money_paid);

private:
      string Last_Name;
      string First_Name;
      int Age;
      double GPA;
      double Bill_Balance;
      string phone;
      string email;
      string club;
      string course1, course2, course3, course4, course5;
};

student::student(string LN, string FN, int A, double GPA_passed, string P,
string E)
{
Last_Name = LN;
First_Name = FN;
Age = A;
GPA = GPA_passed;
Bill_Balance = 0.0;
phone = P;
email = E;
club = " ";
course1 = " ";
course2 = " ";
course3 = " ";
course4 = " ";
course5 = " ";
}

void Join_Club(string club_joined)
{
 club = club_joined;
}
void Register_for_class(string class_registered_for)
{
if (course1 == " ")
```

```
    course1 = class_registered_for;
else
if (course2 == " ")
    course2 = class_registered_for;
else
if (course3 == " ")
    course3 = class_registered_for;
else
if (course4 == " ")
    course4 = class_registered_for;
else
if (course5 == " ")
    course5 = class_registered_for;
}
void Pay_Bill(double money_paid)
{
  Bill_Balance = Bill_Balance - money_paid;
}

int main()
{
      student KMM("Mattison", "Karen", 29, 4.00, "(856) 299-
                                          2100", "mattison@salemcc.org");
        KMM.Join_Club("Student Govt.");
        KMM.Register_for_class("Introduction to C++");
        KMM.Pay_Bill(1000.00);
}
```

Chapter 5: Functions and Parameters Prerequisites:
Chapters 1, 2, 3, and 4 Goals:

1. Get students writing free (non-member) functions
2. Appreciate why software is structured into functions communicating via argument/parameter associations and returns
3. Revisit the importance of testing
4. Use test drivers to help implement and test free functions
5. Get a first exposure to scope rules while accepting that this is a subject that will be revisited in upcoming chapters

Experiences: I usually present this chapter in three 50-minute lectures using the presentations available with this instructor's manual. Students can implement functions and usually do well on tests with questions that begin like this: "Write a function named foo that...." It helps to let students know that many programming projects include test drivers and to stress that they exist only for testing purposes. These functions are meant to be parts of larger systems and they should be thoroughly tested before they become part

```
// Programmer name:
// Current date:
// Brief Desc:     Section 5.1   Ex. free function
// Free functions are free of a class.


#include
#include
#include
#include "compfun"

void function_One(int X)
{
// Pre-condition:    X will be a whole number
// Post-condition:   X will be displayed

   cout << X << endl;
   X = function_Two(X);
   cout << X << endl;
}

int function_Two(int Y)
{
// Pre-condition:   Y is a whole number
// Post-condition:  Y is returned after multiplying it
//                  by 2

// Some functions only return a value
   return Y * 2;
}

int main()
{
  int hold_value = 55;

  function_One(hold_value);

  return 0;
}
```

```
// Programmer name:
// Current date:
// Brief Desc:     Section 5.3   Ex. identifier scope


#include
#include
#include
#include "compfun"

int Z = 99;         //  This is an example of a
                    //  global identifier

void function_One(int X)
{
// Pre-condition:    X will be a whole number
// Post-condition:   X will be displayed

// Bother identifier X  and  identifier Y are local to
// this function.

   int Y;

   cout << X << " " << Y << endl;  //  this is correct!

   cout << hold_value << endl;     //  this is not correct!

   cout << Z << endl;              //  correct!

}

int main()
{
  int hold_value = 55;           // hold_value is local
                           // to this function

  function_One(hold_value);

  cout << hold_value << endl;   // correct!
  cout << X << " " << Y << endl;// not correct!

  cout << Z << endl;            // correct!

  return 0;
}
```

```
// Programmer name:
// Current date:
// Brief Desc:     Section 5.4   Ex. call by value

#include
#include
#include
#include "compfun"

void function_One(int X)
{
// Pre-condition:    X will be a whole number
// Post-condition:   X will be displayed

   cout << X << endl;
}



int main()
{
  function_One(888);   // the value 888 is passed_by_value to function_One

  return 0;
}
```

```
// Programmer name:
// Current date:
// Brief Desc:     Section 5.4   Ex. call by value

#include
#include
#include
#include "compfun"

void function_One(int & X)               // notice the & character
{                                  // X is now alais for identifier
                                   // in main function

// Pre-condition:    X will be a whole number
// Post-condition:   X will be displayed

   cout << X << endl;
   X = 444;

}



int main()
{
  int hold_value = 888;
  function_One(hold_value);  // the value 888 is passed_by_reference to
                          //     function_One via hold_value

  cout << hold_value << endl;  // when hold_value is displayed is will
                               // display 444 not 888;
  return 0;
}
```

```
// Programmer:
// Current Date:
// Brief Desc:

#include
#include
#include
#include "compfun"
#include "bankAccount.h"

using namespace std;

int main()
{
      bankAccount KMM("Karen's", 1000.00);

      bankAccount KMM1;

      return 0;
}
```

```
#include
#include
#include
#include "compfun"

class bankAccount {
public:
   // constructor follows

   bankAccount(string acctName, double initBal);
   bankAccount(void);
   void withdraw(double money);
   void deposit(double money);
   double balance(void);
   string name();

private:
      string my_name;
      double my_balance;
};

bankAccount::bankAccount(void)
{
      // post-condition:
  my_name = " ";
  my_balance = 0.0;

}

bankAccount::bankAccount(string acctName, double initBal)
{
      // pre-condition:
      // post-condition:
  my_name = acctName;
  my_balance = initBal;

}

void bankAccount::withdraw(double money)
{
      // pre-condition:
      // post-condition:
  my_balance = my_balance - money;

}

void bankAccount::deposit(double money)
{
      // pre-condition:
      // post-condition:
  my_balance = my_balance + money;
}
```

```
double bankAccount::balance(void)
{
      // post-condition:

    return my_balance;
}

string bankAccount::name(void)
{

      // post-condition:
    return my_name;
}
```

```
//Programmer Name:
//Date:
//Brief Desc:


#include
#include
#include

using namespace std;

class libraryBook
{
public:
      libraryBook(void);
      libraryBook(string initTitle, string initAuthor);
      void borrowBook(string borrowersName);
      int returnBook();
      string borrower() const;
private:
      string my_author;
      string my_title;
      string my_borrower;
};
libraryBook::libraryBook(void)
{
      // pre-condition:    n/a
      // post-condition:   books state is set to null
      my_author = " ";
      my_title = " ";
      my_borrower = " ";
}
libraryBook::libraryBook(string initTitle, string initAuthor)
{
      // pre-condition:   2 strings will be sent.  1st is title
      //                  and 2nd is author
      // post-condition:  book's state is updated

      my_author = initAuthor;
      my_title = initTitle;
      my_borrower = " ";

}
void libraryBook::borrowBook(string borrowersName)
{
      // pre-condition:  must pass name of borrower as string
      // post-condition:  book's borrower will be updated

      my_borrower = borrowersName;

}
int libraryBook::returnBook()
```

```
{
      // pre-condition:          n/a
      // post-condition:         book's borrower is reset to null

      my_borrower = " ";
      return 0;
}
string libraryBook::borrower() const
{
      // pre-condition:          n/a
      // post-condition:       returns the name book borrower

      return my_borrower;
}



int main( )
{

  libraryBook book1("Cat in the hat", "Dr. Suess");

  cout << book1.borrower() << endl;

  book1.borrowBook("karen mattison");

  cout << book1.borrower() << endl;

  book1.returnBook();

  cout << book1.borrower() << endl;

       return 0;

}
```

```
class room {
public:

room(string nameOfRoom, int roomTemp);
void setDesiredTemp(int temp);
int getDesiredTemp(void);

private:
 string my_roomName;
 int my_temp;

};

room::room(string nameOfRoom, int roomTemp)
{
// pre-condition:
// post-condition:
   my_roomName = nameOfRoom;
   my_temp = roomTemp;

}
void room::setDesiredTemp(int temp)
{
// pre-condition:
// post-condition:
      my_temp = temp;

}
int room::getDesiredTemp(void)
{
// pre-condition:
// post-condition:
   return my_temp;
}



int main()
{
  room classroom("classroom S106", 75);
  return 0;
}
```

Chapter 7: Selection Prerequisites: Chapters 1, 2, 3, 4, and 5 (if you skip 6, don't assign projects marked with a 6 prerequisite) Goals: 1. Learn selection control 2. Consider more algorithmic patterns such as guarded action and alternate selection. 3. Implement algorithms using if, if…else, and nested if…else 4. See multiple returns 5. Practice branch and boundary testing and use larger test drivers. Experiences: I usually present this chapter in four or five 50-minute lectures using the presentations available with this instructor's manual. The if...else statement is an easy topic for students. I usually just mention the switch statement as an excuse to introduce the integral type char. It seems like most people either really like switch or don't use it at all. It is a bit awkward to talk about nested logic but then later provides the option of multiple returns from a function. So we can get code like this even though we don't need the else's string letterGrade(double percentage) { if(percentage >= 93) return "A"; else if(percentage >= 90) return "A-"; else if(percentage >= 87) return "B+"; else if(percentage >= 83) return "B" // ... } Actually, students had no problems with multiple returns. Instead some students still have problems with the function heading (parameters) and placement of functions in the proper location in the file (in relation to the main function). So in this second edition I have supplied function headings in some of the first programming projects. The idea of writing parameters caused trouble for some. Others placed the heading in strange places and the body elsewhere. Students like to put a semicolon at the end of the heading and then have trouble trying to figure out the error messages in code like this double max(double x, double y); { //... } Have students review the Chapter 5 and 6 Programming Tips beginning on page 165 and 226, respectively. Using the familiar classes of grid and bankAccount to introduce new concepts saved the overhead of introducing new classes with new concepts.

```
//Programmer Name:
//Date:
//Brief Desc:


#include
#include
#include
#include "compfun"

using namespace std;

char DisplayMainMenu(void)
{
// pre-condition   N/A
// post-condition  displays the menu

    char ProceedInd;

      cout << "\n\n\n\n\n \t\t\t\t\t  Payroll Calc System";

      cout << "\n\n \t\t\t  Proceed  ( Y or N ): ";
      cin >> ProceedInd;

      return ProceedInd;

}


void DisplayTerminateMenu(void)
{
    cout << "\n\n\n\n\n \t\t\t\t\t Processing Complete";
}

void DisplayProcessMenu(void)
{

}

int main( )
{

    char value_hold;

    value_hold = DisplayMainMenu();

      if ( value_hold == 'Y' )
      {
            DisplayProcessMenu();
      }
      else
      {
```

```
                DisplayTerminateMenu();
        }


        return 0;

}
```

```
// Programmer Name:        K.M.Mattison
// Current Date:           4/23/02
// Brief Description:   p288, 7J

#include
#include
#include
#include "compfun"


using namespace std;


string grade(double pass_grade)
{

      if (pass_grade > 93.0)
      {
            return "A";
      }
      else
      if (pass_grade >= 90.0  &&  pass_grade <= 93.0 )
      {
            return "A-";
      }
      else
      if (pass_grade >= 87.0  &&  pass_grade <= 90.0 )
      {
            return "B+";
      }
      else
      if (pass_grade >= 83.0  &&  pass_grade <= 87.0 )
      {
            return "B";
      }
      else
      if (pass_grade >= 80.0  &&  pass_grade <= 83.0 )
      {
            return "B-";
      }
      else
      if (pass_grade >= 77.0  &&  pass_grade <= 80.0 )
      {
            return "C+";
      }
      else
      if (pass_grade >= 70.0  &&  pass_grade <= 77.0 )
      {
            return "C";
      }
      else
      if (pass_grade >= 60.0  &&  pass_grade <= 70.0 )
```

```
        {
                return "D";
        }
        else
        if (pass_grade < 60.0 )
        {
                return "F";
        }

}

int main()
{
     double grade_hold;
        string letter_grade;

TopOfLoop:

        cout << "Enter the test/quize grade or -1 to quit: ";
        cin >> grade_hold;


        if (grade_hold < 0)
        {
                goto EndProgram;
        }

        letter_grade = grade(grade_hold);
        cout << "Your grade is : " << letter_grade << endl;

        goto TopOfLoop;

EndProgram:

        return 0;

}
```

```
// Programmer Name:       K.M.Mattison
// Current Date:          4/23/02
// Brief Description:    Example of a switch structure being
//                         implemented

#include
#include
#include
#include "compfun"

using namespace std;


void grade(char grade_passed)
{

      switch(grade_passed)
      {

      case 'A':
            cout << "Your grade is > 90.0 " << endl;
            break;
      case 'B':
            cout << "Your grade is > 80.0 " << endl;
            break;
      case 'C':
            cout << "Your grade is > 70.0 " << endl;
            break;
      case 'D':
            cout << "Your grade is > 60.0 " << endl;
            break;
      case 'F':
            cout << "your grade is < 60.0 " << endl;
            break;
      }




}







int main()
{
      char letter_grade;

    do {
        cout << "Enter your test/quize grade OR enter Q(quit): " ;
        cin >> letter_grade;
        grade(letter_grade);


        } while(letter_grade != 'Q' );

        return 0;

}
```

## Appendix D

| Enhanced DE | *Student 1* | *Student 2* | *Student 3* | *Student 4* | *Student 5* | *Student 6* | *Student 7* | *AVG* | *HIGH* | *LOW* | *Mediam* | *Mode* | *Std Dev* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ATT | 25 | 30 | 34 | 33 | 32 | 39 | 37 | 33 | 39 | 25 | 33 | #N/A | 4.6 |
| MOT | 24 | 27 | 26 | 25 | 25 | 38 | 31 | 28 | 38 | 24 | 26 | 25 | 5.0 |
| TMT | 22 | 19 | 21 | 18 | 17 | 33 | 27 | 22 | 33 | 17 | 21 | #N/A | 5.7 |
| ANX | 29 | 18 | 29 | 26 | 36 | 36 | 28 | 29 | 36 | 18 | 29 | 29 | 6.2 |
| CON | 10 | 25 | 25 | 26 | 25 | 38 | 37 | 27 | 38 | 10 | 25 | 25 | 9.3 |
| INP | 28 | 29 | 26 | 15 | 31 | 37 | 28 | 28 | 37 | 15 | 28 | 28 | 6.6 |
| SMI | 14 | 22 | 23 | 18 | 17 | 25 | 21 | 20 | 25 | 14 | 21 | #N/A | 3.8 |
| STA | 24 | 15 | 21 | 24 | 17 | 27 | 15 | 20 | 27 | 15 | 21 | 24 | 4.8 |
| SFT | 20 | 14 | 22 | 24 | 23 | 31 | 27 | 23 | 31 | 14 | 23 | #N/A | 5.4 |
| TST | 25 | 34 | 38 | 34 | 33 | 38 | 36 | 34 | 38 | 25 | 34 | 34 | 4.4 |

ᵓf stᵘ

Er

| **Enhanced DE** | *Student 3* | *Student 6* | *Student 7* | *Average* | *High* | *Low* | *Mode* | *Mediam* | *Std Dev* |
|---|---|---|---|---|---|---|---|---|---|
| ATT | 34 | 39 | 37 | 37 | 39 | 34 | #N/A | 37 | 3 |
| MOT | 26 | 38 | 31 | 32 | 38 | 26 | #N/A | 31 | 6 |
| TMT | 21 | 33 | 27 | 27 | 33 | 21 | #N/A | 27 | 6 |
| ANX | 29 | 36 | 28 | 31 | 36 | 28 | #N/A | 29 | 4 |
| CON | 25 | 38 | 37 | 33 | 38 | 25 | #N/A | 37 | 7 |
| INP | 26 | 37 | 28 | 30 | 37 | 26 | #N/A | 28 | 6 |
| SMI | 23 | 25 | 21 | 23 | 25 | 21 | #N/A | 23 | 2 |
| STA | 21 | 27 | 15 | 21 | 27 | 15 | #N/A | 21 | 6 |
| SFT | 22 | 31 | 27 | 27 | 31 | 22 | #N/A | 27 | 5 |
| TST | 38 | 38 | 36 | 37 | 38 | 36 | 38 | 38 | 1 |

| **DistanceEd** | *Student 1* | *Student 2* | *Student 3* | *Student 4* | *Student 5* | *Student 6* | *Student 7* | *AVG* | *HIGH* | *LOW* | *Mediam* | *Mode* | *Std Dev* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *ATT* | 34 | 38 | 33 | 26 | 38 | 34 | 38 | 34 | 38 | 26 | 34 | 38 | 4.3 |
| *MOT* | 32 | 34 | 24 | 29 | 35 | 34 | 38 | 32 | 38 | 24 | 34 | 34 | 4.6 |
| *TMT* | 27 | 33 | 17 | 18 | 29 | 16 | 33 | 25 | 33 | 16 | 27 | 33 | 7.5 |
| *ANX* | 25 | 32 | 25 | 20 | 25 | 18 | 28 | 25 | 32 | 18 | 25 | 25 | 4.7 |
| *CON* | 24 | 37 | 26 | 25 | 32 | 24 | 31 | 28 | 37 | 24 | 26 | 24 | 5.0 |
| *INP* | 33 | 27 | 25 | 30 | 26 | 25 | 29 | 28 | 33 | 25 | 27 | 25 | 3.0 |
| *SMI* | 18 | 24 | 19 | 13 | 22 | 20 | 21 | 20 | 24 | 13 | 20 | #N/A | 3.5 |
| *STA* | 20 | 17 | 21 | 24 | 23 | 30 | 21 | 22 | 30 | 17 | 21 | 21 | 4.1 |
| *SFT* | 25 | 20 | 24 | 26 | 28 | 30 | 25 | 25 | 30 | 20 | 25 | 25 | 3.2 |
| *TST* | 29 | 37 | 28 | 29 | 31 | 27 | 30 | 30 | 37 | 27 | 29 | 29 | 3.3 |

**This data represents just the high performers in each group**

| **DistanceEd** | *Student 4* | *Student 5* | *Student 6* | *Average* | *High* | *Low* | *Mode* | *Mediam* | *Std Dev* |
|---|---|---|---|---|---|---|---|---|---|
| *ATT* | 26 | 38 | 34 | 33 | 38 | 26 | #N/A | 34 | 6 |
| *MOT* | 29 | 35 | 34 | 33 | 35 | 29 | #N/A | 34 | 3 |
| *TMT* | 18 | 29 | 16 | 21 | 29 | 16 | #N/A | 18 | 7 |
| *ANX* | 20 | 25 | 18 | 21 | 25 | 18 | #N/A | 20 | 4 |
| *CON* | 25 | 32 | 24 | 27 | 32 | 24 | #N/A | 25 | 4 |
| *INP* | 30 | 26 | 25 | 27 | 30 | 25 | #N/A | 26 | 3 |
| *SMI* | 13 | 22 | 20 | 18 | 22 | 13 | #N/A | 20 | 5 |
| *STA* | 24 | 23 | 30 | 26 | 30 | 23 | #N/A | 24 | 4 |
| *SFT* | 26 | 28 | 30 | 28 | 30 | 26 | #N/A | 28 | 2 |
| *TST* | 29 | 31 | 27 | 29 | 31 | 27 | #N/A | 29 | 2 |

| **NonDistance** | *Student 1* | *Student 2* | *Student 3* | *Student 4* | *Student 5* | *Student 6* | *Student 7* | *Student 8* | *Student 9* | *AVG* | *HIGH* | *LOW* | *Mediam* | *Mode* | *Std Dev* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *ATT* | 32 | 34 | 29 | 37 | 33 | 28 | 35 | 21 | 30 | 28 | 37 | 21 | 32 | #N/A | 4.7 |
| *MOT* | 24 | 35 | 31 | 31 | 32 | 34 | 37 | 26 | 39 | 29 | 39 | 24 | 32 | 31 | 4.9 |
| *TMT* | 24 | 22 | 20 | 15 | 21 | 28 | 30 | 27 | 37 | 22 | 37 | 15 | 24 | #N/A | 6.5 |
| *ANX* | 6 | 25 | 23 | 20 | 20 | 23 | 22 | 38 | 16 | 19 | 38 | 6 | 22 | 23 | 8.4 |
| *CON* | 17 | 27 | 25 | 15 | 22 | 32 | 28 | 26 | 28 | 22 | 32 | 15 | 26 | 28 | 5.5 |
| *INP* | 23 | 21 | 24 | 29 | 34 | 31 | 28 | 21 | 36 | 25 | 36 | 21 | 28 | 21 | 5.5 |
| *SMI* | 14 | 21 | 16 | 14 | 26 | 12 | 23 | 21 | 15 | 16 | 26 | 12 | 16 | 14 | 4.8 |
| *STA* | 20 | 16 | 16 | 21 | 29 | 22 | 27 | 16 | 33 | 20 | 33 | 16 | 21 | 16 | 6.2 |
| *SFT* | 17 | 29 | 22 | 21 | 34 | 24 | 31 | 15 | 28 | 22 | 34 | 15 | 24 | #N/A | 6.4 |
| *TST* | 24 | 26 | 24 | 21 | 24 | 24 | 33 | 36 | 26 | 24 | 36 | 21 | 24 | 24 | 4.9 |

| **NonDistance** | *Student 1* | *Student 4* | *Student 5* | *Student 6* | *Average* | *High* | *Low* | *Mode* | *Mediam* | *Std Dev* |
|---|---|---|---|---|---|---|---|---|---|---|
| *ATT* | 32 | 37 | 33 | 28 | 33 | 37 | 28 | #N/A | 33 | 4 |
| *MOT* | 24 | 31 | 32 | 34 | 30 | 34 | 24 | #N/A | 32 | 4 |
| *TMT* | 24 | 15 | 21 | 28 | 22 | 28 | 15 | #N/A | 23 | 5 |
| *ANX* | 6 | 20 | 20 | 23 | 17 | 23 | 6 | 20 | 20 | 8 |
| *CON* | 17 | 15 | 22 | 32 | 22 | 32 | 15 | #N/A | 20 | 8 |
| *INP* | 23 | 29 | 34 | 31 | 29 | 34 | 23 | #N/A | 30 | 5 |
| *SMI* | 14 | 14 | 26 | 12 | 17 | 26 | 12 | 14 | 14 | 6 |
| *STA* | 20 | 21 | 29 | 22 | 23 | 29 | 20 | #N/A | 22 | 4 |
| *SFT* | 17 | 21 | 34 | 24 | 24 | 34 | 17 | #N/A | 23 | 7 |
| *TST* | 24 | 21 | 24 | 24 | 23 | 24 | 21 | 24 | 24 | 2 |

## Appendix E

**CS121 Distance Learning Questionnaire** **Student Name:** ______________________________

Karen Mattison, Instructor, Computer Science, is investigating the benefit of having a WebCt based course web site, used in connection with or as a Distance Learning effort, for any Computer Science course offered here at Salem Community College. Feedback concerning your experiences with the Introduction to C++ course's WebCt-based web site during the Spring/2002 semester will help to further substantiate any quantitative-based results that are determined. It would be greatly appreciated if you could complete the following questionnaire and return it by the next class meeting. If you have any questions, please call me at your convenience at (856) 299-2100.

*Please answer each question by checking the appropriate box.*

1. Did you have prior computer programming experience before taking this course?

   [ ] YES [ ] NO

2. What experience level would you place yourself at based on your prior programming experience?

   [ ] Novice/Beginner [ ] Intermediate [ ] Advanced

3. Did you have prior PC experience before taking this course?

   [ ] YES [ ] NO

4. What experience level would you place yourself at based on your prior PC experience?

   [ ] Novice/Beginner [ ] Intermediate [ ] Advanced

5. Which section were you registered for during the Spring/2002 semester of CS121?

   [ ] Day [ ] Evening [ ] Distance Education

6. What percentage of your time had you spent using the WebCt-based course web site for CS121 Intro. To C++ during the Spring/2002 semester?

| | | |
|---|---|---|
| [ ] 0 % | [ ] 26% - 50% | [ ] 76% or more |
| [ ] 1% - 25% | [ ] 51% - 75% | |

7. If your answer to question 1 was not 0% then identify which items available on this site that you did use.

| | | |
|---|---|---|
| [ ] Calendar | [ ] Discussion Board | [ ] Syllabus |
| [ ] Class Notes | [ ] Testing/Assessment | [ ] email |

8. Did the availability of the web site provide any benefit to you?

   [ ] None [ ] Some [ ] Much

   Note; please comment how. ______________________________________________

   ________________________________________________________________

   ________________________________________________________________

   ________________________________________________________________

9. Based on your experience with this course, would you benefit in any way if any/all SCC computer science courses provided a WebCt-based course web site?

   [ ] YES [ ] NO

   Note; please comment how. ______________________________________________

   ________________________________________________________________

10. Please provide any additional comments, positive or negative, regarding your experiences with CS121 Introduction to C++ during the Spring/2002 semester on the reverse side of this document.