# CSC 111 – Fundamentals of Computer Science

**Course Description:** (4h) Lecture and laboratory. Introduction to the basic concepts of computer programming and algorithmic problem solving for students with little or no programming experience. Recommended as the first course for students considering a major or minor in computer science; also appropriate for students who want computing experience applicable to other disciplines.

**Professor:** Dr. William Turkett - Manchester 240, 758-4427, email: turketwh@wfu.edu
**Office Hours:** Tuesday 3:30pm-5pm, Wednesday 10am-12noon, Friday 10am-11:30am
**Teaching Assistant**: TBA

**Meeting Times:** 11:00-12:15 TR, Manchester 241 – lecture
3:30-4:45 Thursday, Manchester 241 – lab

**Webpage:** Material from the class will be uploaded into Sakai under the Resources tab. Please check Sakai frequently for updates concerning the class.

**Textbook:** Starting Out With Visual Basic 2012 by Gaddis and Irvine - Pearson – ISBN-10: 013312808-3

**Grading:**

- Tests – 40% (2 in-class and 1 final – equally weighted)
- Labs – 30% (13 planned)
- Homeworks – 25%
- Reading Quizzes - 5%

A failing grade in the lab component of the class leads to a failure for the whole class, regardless of the remaining grades.

**Expected Grading Scale:**

| | | |
|---|---|---|
| A 93-100 | A- 90-92 | B+ 87-89 |
| B 83-86 | B- 80-82 | C+ 77-79 |
| C 73-76 | C- 70-72 | D+ 67-69 |
| D 63-66 | D- 60-62 | F 0-59 |

An *A* should represent mastery of the subject material, a *B* advanced understanding, a *C* basic understanding, a *D* minimal understanding, and *F* a failure in understanding.

**Tests and Final Exam:**
There will be three closed-book tests (including the final) in the course. These tests may include material from the textbook, lectures, homeworks, and programming assignments and may include cumulative aspects. Make up tests will be allowed only if the absence is excused by the University.

**Labs:**
Lab will be assigned weekly during the semester and will be submitted via Sakai. A grading rubric for each lab will be provided for each lab. Labs can be submitted up to one day late, but at a cost of one letter grade (10 points) for that day. After that, you may submit the assignment via email for feedback, but a grade of zero will be awarded. Ideally, given the way the class is designed, you will be able to complete or nearly complete the labs during the scheduled lab times.

**Homeworks:**
Homework to reinforce the material discussed will be assigned regularly during the course of the semester, to be submitted via Sakai. These may come in different forms, including problems to hand in, small programs to analyze or write, or Sakai-based assignments (multiple choice/fill in the blank/etc). Homeworks will have a due-date controlled by Sakai and can't be submitted late.

**Reading Quizzes:**
Before most classes, you will be asked to read a section from the book, read an article online, and/or view a video online and then to answer a few questions about the reading/viewing in Sakai. These quizzes are a way for you to provide me feedback on material you would like more help on. They will be due at midnight the evening before class and can't be submitted late.

**Work Expectations:**
The work expectation model at Wake Forest, and at many colleges, is for each 1 hour in class, 2 hours of work outside of class. Under this model, the standard 15-credit hour semester is akin to a 45-hour work week. Since this is a 4-credit hour course, you should expect on average, 8 hours of work outside of class each week.

**Academic Integrity:**
Unless otherwise explicitly specified (there will be some cases where collaboration is OK), all work should be done independently by each student. Copying of partial or complete work will result in a zero for that assignment and be referred to the University Judicial System. You should keep evidence when possible to demonstrate your own work. Should a question of authorship arise you will be expected to produce documents that trace the development of your work.

**Learning Assistance:**
If you have a disability that may require an accommodation for taking this course, please contact the Learning Assistance Center (758-5929) within the first two weeks of the semester.

**Topics Covered:**
In this course, we will cover several fundamental topic areas. One of the most important will be understanding the core constructs that are employed when writing computer programs – primarily *ways of representing data* and *ways of manipulating data with instructions*. Data representation includes how we represent *primitiv*e information, such

as numbers and letters, as well as how we *model the real world*. Another fundamental topic is the idea of an *algorithm* – how to define a rigorous, automatable process to help us solve problems of interest. Several specific types of algorithms will be explored. Mechanisms for *interacting with users of a computer program* will make up another fundamental topic area. Finally, the *mechanics of writing programs* will be studied as we develop the topics discussed above.

**Course Outcomes:**
By the end of this course, students will be able to:

- Analyze and explain the behavior of simple programs involving the fundamental programming constructs.
- Modify and expand short programs that use standard conditional and iterative control structures and functions.
- Design, implement, test, and debug a program that uses the fundamental programming constructs.

- Choose appropriate conditional and iteration constructs for a given programming task.
- Apply the techniques of functional decomposition to break a program into smaller pieces.
- Describe the mechanics of parameter passing.

- Discuss the importance of algorithms in the problem-solving process.
- Identify the necessary properties of good algorithms.
- Create algorithms for solving simple problems.
- Use pseudo-code or a programming language to implement, test, and debug algorithms for solving simple problems
- Describe strategies that are useful in debugging.

- Describe the representation of numeric and character data.
- Understand how precision and round-off can affect numeric calculations
- Discuss the use of primitive data types and built-in data structures.
- Describe common applications for studied data-structures.
- Write programs that use primitive data types

- Describe the concept of recursion and give examples of its use.
- Identify the base case and the general case of a recursively defined problem.
- Describe the divide-and-conquer approach.

- Explain the difference between event-driven programming and command-line programming.
- Design, code, test, and debug simple event-driven programs that respond to user events.
- Develop code that responds to exception (error) conditions raised during execution.

- Justify the philosophy of object-oriented design and the concepts of encapsulation, abstraction, inheritance, and polymorphism
- Design, implement, test, and debug simple programs in an object-oriented programming language.
- Describe how the class mechanism supports encapsulation and information hiding.

*Why should you be interested in this (long) list of outcomes*? First, think of this list as your "computer science resume" – your developed skill set after completing the course. It is a list understandable by anyone in the computer field. Second, these outcomes effectively define the prerequisites for building on in future independent or formalized computer science education. No matter which version of CSC 111 you take, these are the fundamental outcomes everyone will achieve. Finally, and maybe most importantly in the near term, these outcomes are what I will be evaluating you on (on tests, labs, etc…)!

*What is the relationship to business applications?* There are a number of ways this course will focus on business applications of computing. First off, most of the labs and many examples will be oriented towards concepts from business. In particular, I am working to transform accounting problems into practical examples we can employ (I am open to suggestions). We will also be introduced to database systems, large stores of information such as one can encounter in customer tracking (how does Harris Teeter manage all of its VIC card data?). Finally, we will be learning the programming language Visual Basic, one of the most commonly used languages used for developing Windows-based business applications and a language very similar to programming tools embedded in Excel.

**University Closure:**
In the event that the University closes due to a health pandemic or other emergency, you will be provided with my home address, phone number, and a *CSC 111 Lecture Plan* document. You are requested to read the textbook material denoted within that document. Lecture materials, in the form of Powerpoint slides and/or videos; programming exercises; homeworks; and examination materials, will be distributed electronically via email or via postal mail during the closure period. If the Internet is available, you should submit electronic versions of your solutions through the Sakai site, and if that fails, send the electronic solutions of the homeworks and programming exercises to either my WFU email address or turketwh@gmail.com. Tests and quizzes should be taken closed book, without access to papers, persons, or other resources, and submitted via postal mail. A return date for the examinations will be specified in the mailing.

The Department of Computer Science would appreciate your help preparing for emergency situations in which students might be away from campus for an extended period of time during the course of a semester. For example, extreme weather or widespread health concerns might lead to an extended, but temporary, closing of campus facilities. Under such circumstances we would like for you to be able to continue your academic studies through electronic or postal communication channels. Please assist your instructor by providing contact information. This information will remain with your instructor and will not be disseminated in any way. Collected information will be deleted at the end of the semester.