CSC 111 Introduction to Computer Science
Syllabus
Fall 2016
Class meets in Manchester 024: Monday/Wednesday 2:00 – 3:15, Lab meets Wednesday 3:30 – 4:45

Instructor: Don Gage
Office: Manchester 240
Email: gagehd@wfu.edu or dgage@wakehealth.edu
Office hours: (I enjoy meeting with students – individually or in groups)
Scheduled: TBA
  Drop by I will make time if at all possible.
  By appointment I try to be available as needed, so please email.
  Via email Simple and convenient.
TA: Ruidan Li: TBA

Textbook: Sign up at www.zyBooks.com
Enter zyBook code WFUCSC111GageFall2016
Click Subscribe
The cost to subscribe is $67
A video with more detail on subscribing is at https://vimeo.com/135692064

Optional Text: Also, while not required, I would suggest buying the Java 8 Pocket Guide on Amazon. It is a handy reference and can be purchased for as little as $3.29 used.

Course website: There is a Sakai page where handouts and other documents will be distributed.

Grading:
Class attendance, participation, and exercises in the online text 10%
Tests and Quizzes (2 Tests and 1 Final) 60%
Labs 30%

The Final Exam is scheduled for Tuesday, December 13th at 2:00 PM. Please note when making travel plans.

Final grade assignment will adhere minimally to table below. That is, achieving the number grade will ensure that the letter grade will be at least the one shown.

<table>
<thead>
<tr>
<th>Number Grade</th>
<th>Letter Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>92 or above</td>
<td>A</td>
</tr>
<tr>
<td>90 thru 91.99</td>
<td>A – 70 thru 77.99: C</td>
</tr>
<tr>
<td>88 thru 89.99</td>
<td>B + 68 thru 69.99: D +</td>
</tr>
<tr>
<td>82 thru 87.99</td>
<td>B 62 thru 67.99: D</td>
</tr>
<tr>
<td>80 thru 81.99</td>
<td>B – 60 thru 61.99: D –</td>
</tr>
<tr>
<td>78 thru 79.99</td>
<td>C + below 60 %: F</td>
</tr>
</tbody>
</table>
Course Description from Wake Forest Bulletin, 2015-2016:

111. Introduction to Computer Science. (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. Lab – 2 hours. (D)

Course Learning Objectives:

In this course, students will learn the fundamentals of computational problem-solving. They will learn how to write logical sequences of statements that constitute an algorithm. They will develop algorithms and implement them in a particular programming language (in this case, Java). As they write programs, students will learn how to use variables, constants, assignment statements, conditions, loops, input and output statements, arrays, functions, objects, classes, and inheritance. They will learn to write programs that use both textual and graphical user interfaces. Students will also gain experience with an integrated development environment (an IDE) – the programming environment that provides them with an editor, compiler, run-time environment, debugger, and plug-ins. They will learn how to write, compile, and debug programs in the given IDE (in this case IntelliJ). By the end of the semester, students will be able to develop and implement algorithms to solve simple to moderately difficult problems.

Course Topics:

Problem Solving
The concept of an algorithm
Algorithmic problem-solving and problem-solving strategies
Structured decomposition and top-down design
Debugging strategies
Introduction to unit testing

Fundamental Constructs
Basic syntax and semantics of a higher-level language, Java
Variables, types, expressions, and assignment statements
Simple input and output
Conditional and iterative control structures
Recursion
Functions and parameter passing
Scope of identifiers
Call by value vs. call by reference

Data Representation, Data Types, and Data Structures
Binary and hexadecimal base systems
Primitive types such as character, integer, and floating point numbers
Arrays (1D and 2D) and array lists
Strings and string methods
References

Event Driven Programming
Introduction to event-driven programming
Call backs and action listeners
Object-Oriented Programming
Introduction to object-oriented design
Classes and objects and an introduction to inheritance
Encapsulation, data protection, and information-hiding
Separation of interface and implementation

Other Topics
The compile/link/run process
Types of errors:
  - compiler vs. linker
  - syntax vs logical vs run-time
Introduction to exception handling
Introduction to GUI-building
Introduction to memory management (e.g., garbage collection, run-time storage management)
Virtual machine and bytecode interpretation

Tools and Skills:
How to use an IDE effectively, including the debugger
How to use libraries and APIs, including their documentation
File structure fundamentals (directories and subdirectories)

Honor System
Wake Forest is an academic community that subscribes to an honor system. By accepting membership in this community, each student assumes the obligation to be trustworthy in all pursuits. Violations may be referred to the Judicial Council for investigation and determination of appropriate sanctions.

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

Course Policies
- Preparation prior to class is required.
- Class attendance and participation are required.
- Students should bring laptops and power cords to class and lab to be used when specified as part of the class. Other electronic devices (phones, calculators, cameras, recording devices, etc.) are not to be used.
  - Please note that a malfunctioning laptop is not a suitable excuse for missing a lab due date or not taking a quiz. If your laptop is being repaired at The Bridge, you should still be able to do the quiz, lab through the use of a loaner laptop.
- Announcements and other course communications will be made frequently via Sakai messages/email. Students are responsible for monitoring these forums regularly.
- Assignments are due on the due date. If you are unable to turn in an assignment when it is due, please make arrangements to submit it before the due date. Late submissions have consequences. Turn in what you have, even if only partially complete.
- Illness related absences are excusable but if possible please email the instructor if unable to attend class. The instructor retains the right to require documentation justifying absences.
- Other absences may be requested and approved, but please provide a schedule as soon as possible if you have activities that will require you to miss class meetings or reschedule tests/exams or assignment due dates.
Make-up tests/exam may be arranged in the case of unavoidable circumstances such as illness or official university events/activities, etc.

**Plan in the Case of Campus Closing**

Please note the following plan to be followed in the event that the Wake Forest campus is closed for an extended period of time and we are unable to have our regularly-scheduled class meetings.

In normal circumstances, please contact me through my campus email address or campus telephone number.

- **campus email:** gagehd@wfu.edu
- **work phone:** 336-716-5632

In emergency situations or situations where the campus is closed, you may also use the following contacts:

- **alternate email:** dgage@wakehealth.edu
- **work phone:** 336-716-5632

Your course information, including a schedule of assignments, will be posted at on Sakai. After leaving campus, you should regularly consult the schedule on Sakai for updates to the schedule.

Be sure to take your book, computer, and course notes home with you in the event that the campus is closed. We’ll continue with tests and programming assignments, communicating through the internet, email, and/or hard mail.

If internet access is available, assignments should be put into the appropriate Sakai assignment folder.

If the internet is down, I will mail your assignments to you in hard copy, and, by return address, you should mail back a flash drive containing the source code for the implemented program or a DOC file containing the homework solutions.