

# Course Syllabus

## CS-140: Programming with Objects - Fall 2016

**Credits /Contact Hours:** 4 credits, Three 60 minute lectures and one 85-minute lab per week  
Lecture: 2:20pm – 3:20pm (MWF) at Student Wing (SW) 112;  
Labs: Tuesday Afternoon at LNG 103

**Webpage:** [http://www.cs.binghamton.edu/~tbarten1/CS140\\_Fall\\_2016/](http://www.cs.binghamton.edu/~tbarten1/CS140_Fall_2016/)

**Teaching Staff:** Professor: Tom Bartenstein  
Teaching Assistants: Huanxin Zhong  
See web page for contact and office hour details.

**Textbook:** *Big Java Early Objects*, 6th Edition by Cay Horstmann, John Wiley and Sons (2015)  
Java API, Java Tutorial and other information available on-line (see class web page)

**Course Description:** Provides the foundations of software development using Java. Problem solving using object-oriented programming techniques is emphasized. Required laboratory provides supervised problem solving, programming using the command line as well as Eclipse or Netbeans development environments, code backup in a version control repository, debugging and JUnit testing techniques.

**Prerequisites:** CS-120, and CS-100, CS-110 or some familiarity with computers and programming, including:

- Comfort with the concepts outlined in the [Self Assessment](#)
- The ability to use a plain text editor
- Understanding of file systems

CS 120 is usually a prerequisite but can be taken as a co-requisite with the CS Undergraduate Program Director's approval.

**Course Objectives:** Upon completion of this course students will understand and be proficient in the use and application of:

- Programming in Java using both the command line and an IDE
- Declaration, types and assignment of primitive variables in Java.
- A variety of numeric types and their range and precision.
- Control flow constructs: if statements, while loops, for loops, enhanced for loops
- Methods and their parameters, return values. Method calls and arguments.
- Arrays and ArrayLists
- Simple recursive methods
- Lambda Expressions
- I/O for the console. Reading and writing binary and text files
- Event handling and user interaction using GUI interfaces supplied by the instructor.
- Simple methods to draw graphics on the screen using JComponent.
- Interfaces, Classes and Objects, implementation of interfaces, subclasses and inheritance.

- Diagrammatic representation of the run-time structure of objects and the connecting references between them
- Variables declared as reference types and the concept of the run-time type (dynamic type) of a variable
- Javadoc comments
- Overloading and overriding of methods. Polymorphism (dynamic dispatching of method calls)
- Simple JUnit 4.0 testing and basic code debugging
- Exceptions and exception handling

You will also have seen an introduction to the following:

- The call stack and activation records
- Sorting and searching of arrays
- Big-Oh notation
- The definition and use of Java enums
- Java timers for animation
- Java Swing components and layout managers
- Identification of classes and methods in the design of object-oriented software

### **Main Topics:**

- Primitive and reference data types,
- Variables, expressions, assignment,
- Functions/methods, parameters,
- Selection, iteration, recursion,
- Exception handling,
- Generic linear data structures and maps,
- File types, file I/O,
- Simple GUIs,
- Programming to an interface,
- Use of inheritance,
- javadoc documentation
- Introduction to Java threads.

### **Responsibilities:**

**Instructor:** Present correct material in an organized fashion. Help students understand material. Design homework and tests to help the students learn the material. Be available to help students.

**Student:** Attend every class on time, ready to learn and to participate. Stay caught up. Ask questions. Do homework in an honest and timely fashion. Coursework is cumulative. If you don't understand the material at the beginning it will be difficult to catch up later. No one is perfect - please correct us if you find an error.

Many skills are needed to succeed in this course and in the field of Computer Programming. It is important to read the textbook, especially as directed in class. Give yourself plenty of time to work on assignments--leave nothing to the last minute. Attend class and lab. There will be lots of evaluations and every point counts.

**Lecture Notes:** Lecture Notes for each lecture will be posted on the class web-site in PDF format before lectures. *Lecture notes do not substitute for class attendance*, since (i) they will not be complete and (ii) significant parts of lectures, including discussions and in-class exercises, may not come from the class notes.

**Grading:** Your grade will be based on:

In Class Tests	30%
Quizzes and in class activity	5%
Lab Activities	10%
In Lab Tests	15%
Assignments	30%
Project	10%

You need at least a 65 average to pass, but the following criteria also apply:

- You must earn at least a 60 average on the in-class and in lab tests
- You must earn at least a 60 average on the assignments. If you do not do the assignments, you will fail the course

The “total grade” on Blackboard has no relations to the weighted grades above – ignore it.

**Assignments:** Unless otherwise noted, assignments are individual assignments. This means that all work submitted will have been done by you. You may (and are encouraged to) seek help from others, including the instructor, TAs, and classmates. Help includes assistance with:

- using the software tools needed to complete an assignment
- understanding the specifications or requirements of an assignment

Get TA or instructor help with:

- evaluating strategies for solving a problem
- debugging code that you have written
- interpreting compiler and run-time error messages

Obvious violations of the principles of academic honesty include, but are not limited to:

- submitting another person's work (in whole or in part) as your own
- submitting the same work (with or without minor changes such as changes of names and the order of code sections) as another student

Assignments are to be submitted to Blackboard. The filenames and names in the software you submit will be critical. The instructions provided for the assignments and labs must be followed.

Each assignment write-up includes its due date (usually a Wednesday). Due dates may be moved at the last moment if there is a weather emergency, e.g. a snow day. Assignments are due no later than 11:59pm on the due date. Assignments submitted after this deadline will be treated as follows:

- submitted by the following Monday: 20% deducted from the grade
- no assignment will be accepted later unless approval of your course instructor has been obtained!

**Quizzes and Lab Activities:** During classes and labs, there will be graded activities. These will include but are not limited to quizzes and individual or group exercises requiring the reading or writing of Java code. It will not normally be possible to make up quizzes or other lab activities.

**Getting Help:** Please utilize the instructors and CA's office hours for questions and discussion of course related material. Our job is to make you successful, and office hours are a great way to get help. E-mail the instructor or CA's if you need an appointment outside of office hours, or e-mail questions or discussions to the professor directly.

**Academic Honesty Expectations:** Please review the academic honesty document and make sure that you understand it! The link is at: <http://www.binghamton.edu/watson/about/honesty-policy.pdf>. Cheating and copying will NOT be tolerated. For instance, any code turned in will be compared to other students' submissions. If there is significant similarity, even if there are different variable names or comments, all such similar code will receive a zero grade.

**Collaboration:** Students are encouraged to help one another and to form study groups. In Computer Science, you can learn more from your peers than from your instructors and teaching assistants. As long as the help is appropriate, please be generous with your time and expertise when helping fellow students. Doing so is good for you and good for them. You are free to discuss assignments *in general terms* with one another. However, please do not show your work directly to other students. Each student must complete your assignments *individually* (unless indicated otherwise by the instructor). Each of you must write your own code, and you must write up all solutions individually. Students submitting solutions (including code) that are determined to be "too similar" are likely to be punished equally and harshly.

**Academic Stress:** If you are experiencing undue personal or academic stress at any time during the semester or need to talk with someone about a personal problem or situation, I encourage you to seek support as soon as possible. I am available to talk with you about stresses related to your work in my class. Additionally, I can assist you in reaching out to any one of a wide range of campus resources, including:

1. Dean of Students Office: 607-777-2804
2. Decker Student Health Services Center: 607-777-2221
3. University Police: On campus emergency, 911
4. University Counseling Center: 607-777-2772
5. Interpersonal Violence Prevention: 607-777-3062
6. Harpur Advising: 607-777-6305
7. Office of International Student & Scholar Services: 607-777-2510