

CS – 140

Programming with Objects

Instructor: Tom Bartenstein

Course Web Page:

http://www.cs.binghamton.edu/~tbartens/CS140_Fall_2017/

Catalog Description

- Provides the foundations of software development using Java.
- Problem solving using object-oriented programming techniques is emphasized.

CS-240 Course Goals - Java



- Programming in Java
 - command line and IDE
- Java declaration, types, and assignment, including numeric types, ranges, and precision
- Control Flow, if, while, do while, for, and enhanced loops
- Methods, parameters, return values, calls, and arguments
- Defining/Using enums
- Lambda expressions
- Arrays and ArrayLists and Maps
- Recursive methods
- console and File I/O
- Javadoc
- Junit 4.0 testing and debug
- Exceptions and exception handling

Course Goals – Object Orientation

- Object Oriented Design: classes and methods
- Reference type variables and run-time (dynamic) types
- Method overload and override, polymorphism (dynamic dispatch)
- Diagramming run time object structures



Course Goals – OO GUI Design

- Event handling
- User interaction/GUI interfaces
- Screen Graphics using Jcomponent
- Timers for animation
- Java Swing components / Layout managers



Course Goals – Computer Science



- Call Stack and Activation Records
- Sorting/Searching Arrays
- Big-Oh complexity notation

Textbook & Other Resources

- *Big Java Early Objects*, 6th Edition by Cay Horstmann, John Wiley and Sons (2015)
- On line references available through the class web page:
http://www.cs.binghamton.edu/~tbartens/CS140_Fall_2017/

Teaching Staff

- Prof: Tom Bartenstein (tbartens@binghamton.edu)
 - Office Hours: MW 3:30 – 5:30 pm @ EB P-14
 - e-mail
 - By appointment
- Teaching Assistant
 - TBD

Academic Honesty

- For Yourself!
 - College education is not cheap
 - Don't waste it!
- For your peers
 - One person cheating hurts all other students
- For me
 - Cheating wastes time and effort
- For the institution
 - Our reputation depends on you!

Academic Honesty Policy

- Academic dishonesty has no place in a university: It wastes our time and yours, and is unfair to the majority of students.
- The Watson School has an academic honesty code.
- It is your responsibility to read and understand it.
- When in doubt, ask beforehand!

<http://www.binghamton.edu/watson/about/honesty-policy.pdf>

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

Assignments

- 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 following instructions given in Lab. Correct filenames and names in the software you submit will be critical. The instructions provided for the assignments and labs must be followed.

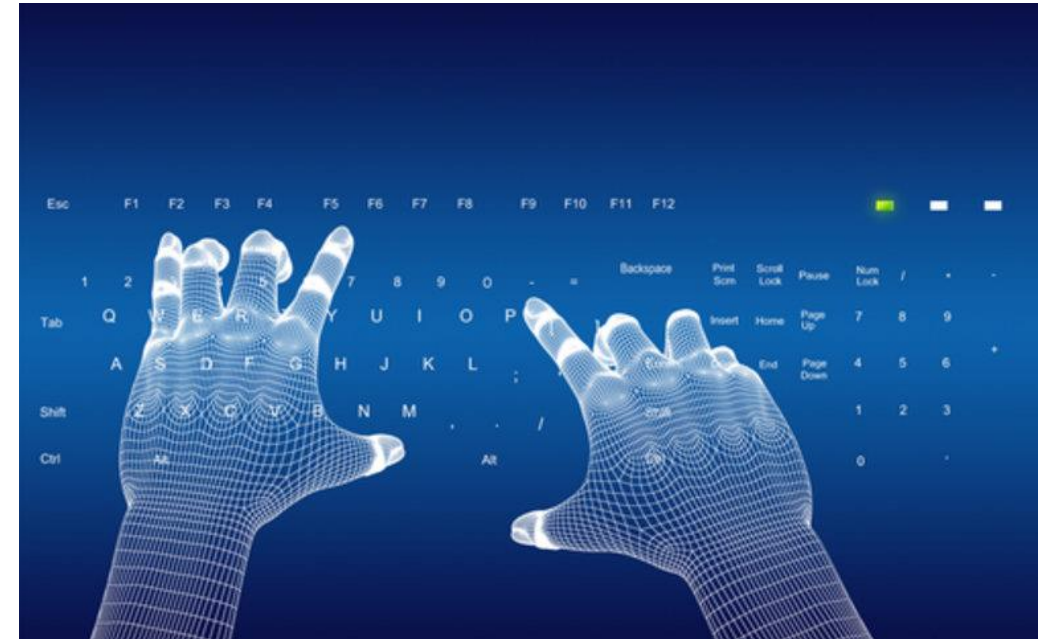
Using your Laptop



- If you want to code on your own laptop, you may do so
 - Alternative: Remote login to “harvey”
- Java is “machine independent” (Java Virtual Machine)
- You must install the Java Development Kit (JDK)
 - Class web page has instructions for Windows and Mac 64 bit
 - Instructions for Unix are on the web
- Eventually, you must install the Eclipse IDE
- Class web page has pointers to get you started
http://www.cs.binghamton.edu/~tbartens/CS140_Fall_2016/javaInstall.html

What is Programming?

- Algorithm
- Language
- Machine Language
- Compilation and Interpretation
- Abstraction vs. Speed
- Generality vs. Efficiency
- Engineering vs. Art vs. Science



How to learn a Language

- You can't teach a language!
- You can teach syntax , grammar, and vocabulary
- You cannot teach style/voice
- You cannot learn to program without practice!
 - You can't read without practice
 - You can't write without practice



Imperative Programming

- The kind of programming we are doing uses the following:
 - We need to work with variables
 - We need to give them values (assignment)
 - We need conditional statements
 - We need loops
 - We need functions
- Imperative: “giving an authoritative command”
- Imperative programming: “Commanding” the computer to solve our problems by writing a computer program

Programming Paradigms

- The main “styles” of programming are called programming paradigms.
- Goals of CS 140
 - Learn procedural programming also called the imperative paradigm
 - Learn about the object oriented paradigm
- Wikipedia lists numerous other paradigms

http://en.wikipedia.org/wiki/Programming_paradigms

What's in a Language?

- Why learn different languages?
 - They are all expressive
- Some languages are easier to code than other languages
 - APL: $x \leftarrow 3\ 4\ \rho\ \iota\ 11$
- Some languages yield code that performs better
- Some languages have really useful libraries
- Trade-offs between complexity and effectiveness

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

Why a Compiled language?

- Why compilers can save money, not just time:

<http://www.businessinsider.com/hhvm-saved-facebook-millions-dollars-2015-7>

- by Drew Parowski – BU Alumni

- Conference Presentation – Why C++ is great for system software

<https://www.youtube.com/watch?v=XqK8Yuoq4ig>