

CSC 250: Introduction to Computer Science via Socially Responsible Design
(proposed title change)
Fall 2009

Instructor: Dr. Ursula Wolz
Tuesday 12:30 PM – 3:20 PM
Friday 12:30 PM – 1:50 PM
Holman Hall 370

Class Website: <http://www.tcnj.edu/~wolz/F09/CSC250F09>
Office hours by appointment via email wolz AT tcnj.edu

A one-semester course in computer science that covers the traditional topics in CS 1 Programming and CS 2 Data Structures. It focuses on software design and how to ethically develop socially responsible computer code.

Content prerequisites: *Exposure* to the concepts and skills of programming, including data, algorithms, heuristics, logic, interaction and abstraction (e.g. synchronous and asynchronous procedures, events, data abstraction (variables, arrays), control structures (loops, selection), arithmetic and logic expression, functions, data input and output.

Content and skills outcomes:

- 1) Demonstrated ability to read, write, analyze and critique in a modern programming language (e.g. Java) including
 - a. Fundamental concepts of programming:
 - i. data organization and abstractions (variables, scope, static and dynamic reference and aggregations),
 - ii. control structures (functions, iteration, recursion, selection),
 - iii. expression evaluation (mathematical, symbolic - string, logical)
 - b. Object-oriented design and implementation principles including, but not limited to: abstraction, inheritance, polymorphism, function overloading, aggregation.
 - c. Data flow and organization including:
 - i. Traditional data structures (lists, stacks, queues, heaps)
 - ii. Synchronous and asynchronous data communication (e.g. events, including but not limited to user input/output, and the use, but not design of asynchronous methods)
- 2) Demonstrated ability to engage in and critique a software design process via implementation in a modern programming language (e.g. Java), including
 - a. design and software life cycle,
 - b. good documentation,
 - c. testing (why and how).
- 3) Demonstrated insight (via writing and coding) of the social impact of computing software, e.g. ability to reason about the ethical ramifications of software design.
- 4) Demonstrated ability to problem solve in community and independently.

Content Resources: There is no assigned text for this course. The class as a whole will construct a wiki that will be an aggregation and synthesis of resources available on the Internet. Organization and content of the wiki will be driven by weekly assignments posed as “Questions to the Class”. Each student will have a clearly identified role each week in updating the wiki. Introduction to the wiki will occur during the first class.

Hardware Resources: You may do all of the coursework on your personal computer, however technical support for the software by Dr. Wolz will only occur in the Mac environment. The software is all free for download. If you do not have a personal computer you can work in either Holman 370 or 123. If you cannot get sufficient access PLEASE let Dr. Wolz know as soon as possible.

Software Resources: are available in the Holman 370. The wiki can be accessed from any browser. Scratch and Processing are both available (and safe) for download.

PBWIKI: <http://tcnjcsc250.pbworks.com/>
SCRATCH: <http://processing.org/>
PROCESSING: <http://scratch.mit.edu/>

Course Structure:

Phase	Learning Goal	Project	Language	# of Weeks	Collaborative
1	Review, refresh knowledge of programming foundations	Design and implement a project of one's own choosing within constraints	Scratch, Processing	2	Independent coding, collaborative critique and debugging.
2	Develop appreciation for OO design via design of a large simulation	Design and implement an elevator simulations	Processing	4	Collaborative design, implementation and critique. Small units assigned for implementation.
3	Modify and extend existing code, with redesign of data structures and objects – move from prototype to production environment	Modify and enhance “Worm World” prototype. Re-implement as application	Processing AND command line Java	4	Collaborative analysis, individual coding responsibilities
4	Independently design a component of a highly interactive project.	Green Games: an Online Game Center that lets people explore “Green” issues.	Processing OR Java	4	Each game is designed and implemented independently, critiqued and debugged as a group, inter-game interface is designed and implemented as a group.

Requirements in temporal order: (percent of final grade)

- 1) (10%) Completion of the “Scratch” project including: design proposal, summary of usability critique by peers, final version, summary of code critique by peers. (week 1)
- 2) (10%) Completion of the “Processing” project (same criteria, week 2)
- 3) (10 %) Construction of the class wiki: (throughout the semester)
 - a. Contributing to the resource pages development. (There is no textbook for this class). Based on questions posed each week, the class will develop an *informative* document that answers the questions and provides links to resources on the internet.
 - b. Contributing to the design documents, testing suites and final documentation of the systems in phases 2 - 4
- 4) (30%) Contributing code development, including coding, debugging, critique on each of the projects in phases 2 – 4 (throughout the semester)
- 5) (10%) Passing grade on five short quizzes (take when ready, may be retaken until two weeks prior to last day of classes):
 - a. Programming fundamentals – write code in Java on demand (1 quiz)
 - b. Traditional data structures - stack, queue, table, heap: (2 quizzes)
 - c. Object oriented implementation with inheritance and polymorphism: (2 quizzes)
- 6) (30%) Final exam – reflective essays on content – no coding.

Grading Breakdown:

Project design and coding (50%), Expository reflection (40%), formal tests on coding (10%)