

Introduction to Interactive Computing
(CSC 101, IMM 120)
Course Requirements, Fall 2009
Tuesday, Friday 10:00 AM – 11:20 AM
Holman 370

Instructor: Dr. Ursula Wolz, Holman Hall 221, tel. 771 2766 , email: wolz@tcnj.edu

Office Hours: by appointment, arranged via email

Course Summary: A first course in computing languages for interactive multimedia. Students are introduced to the art of programming through state-of-the-art multimedia technologies.

Prerequisites: none.

Content and skills outcomes:

1. Demonstrated ability to read, write, analyze and critique basic fundamentals of programming including data, algorithms, heuristics, logic, interaction and abstraction.
2. Able to describe in English and identify, but not write in code:
 - a. the difference between a synchronous and asynchronous procedures,
 - b. events,
 - c. data abstraction (variables, arrays, static and dynamic reference),
 - d. control structures (loops, selection),
 - e. arithmetic and logic expression,
 - f. functions,
 - g. data input and output.
3. Demonstrated ability to engage in a simple version of the software design cycle including project proposal, usability review, code critique and summary report writing.
4. Demonstrated understanding of media data types and procedures including bitmap and vector graphics, and sound, and simple animation and sound production techniques.
5. Demonstrated ability to problem solve in community.
6. Demonstrated ability (via expository writing) to critique the social and ethical ramifications of interactive computing.

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://scratch.mit.edu/>

PROCESSING: <http://processing.org/>

HTML (recommended not required): <http://www.htmlcodetutorial.com/>

Attendance: is required. Reasonable absences will be excused if arranged in advance, or, in case of last minute emergency reported within 24 hours of the missed session (via email or phone). It is expected that you will put in at least 5 hours per week outside of class. You need to engage in the work in this class from the start, keep up with the reading, writing and especially the programming assignments. If you find yourself falling behind, make an appointment to see Dr. Wolz immediately.

Course Structure:

Phase	Theme	Focus	Language	# of Weeks	Project Work
1	Introduction to Interactive Programming via Scratch	Via support from your peers become fluent in a syntax-free programming language	Scratch,	3	A Scratch program of your own design
2	Transition to Processing	Via support from your peers become conversant in a syntax heavy programming language	Processing	2	A very tiny “remix” of an existing project.
3	Programming media: graphics, sound, color, typography, 3D	Learn the foundations of how media is implemented as computing processes and data.	Processing and Scratch	5	A procedural animation implemented in both Scratch & Processing
4	Programming interactivity: events, classes and synchrony.	Learn the foundations of interactivity and object organization.	Processing and Scratch	4	An interactive program implemented in both languages

Requirements: (percent of final grade)

- 1) (40%) Completion of four independent programming projects including design proposal, summary of usability critique by peers, final version, summary of code critique by peers.
- 2) (10%) Construction of the class wiki. Contributing to the questions posed 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.
- 3) (5%) Volunteering to provide a class resource page on the wiki on some topic relevant to the course. This page should be of use to the class in supporting their work. It requires becomes expert in one small topic related to interactive computing.
- 4) (10%) Passing grade on four short quizzes (take when ready, may be retaken until two weeks prior to last day of classes):
 - a. Programming fundamentals in Scratch
 - b. Programming fundamentals in Processing
 - c. Media data manipulation (choice of graphics or sound)
 - d. Hyperlink references
- 5) (5%) Publication of a web-based (HTML) portfolio of projects developed for this class.
- 6) (30%) Final exam – reflective essays on content – no coding.

Grading Breakdown:

Project design and coding (45%), Expository reflection (45%), formal tests on coding (10%)