CMPT 166 - D100 An Animated Introduction to Programming

Instructor(s): Karol Swietlicki

Calendar Objective/Description:
An informal introduction to programming using examples drawn from animation and graphics. Fundamental programming language features are covered, including variables, expressions, statements, loops, functions, and objects. Class design, event-driven programming or other advanced programming techniques may be introduced as needed. No prior programming experience is assumed.

Instructor’s Objectives:
This course is an introduction to programming that teaches programming by exploring many fun and interesting examples chosen from computer 2-D graphics, animations, and games. The course starts with the very basics of getting objects to move in straight lines and bounce around the screen, and works it way all the way up to creating simple games and interactive demos.

Prerequisites:
Recommended: BC Math 12 or equivalent. Students with credit for CMPT 102, 120, 128 or 130 may not take this course for further credit. Students who have taken CMPT 125, 129 or 135 first may not then take this course for further credit.

Quantitative/Breadth-Science

Topics:
- Introduction to programming and the Processing language.
- Coordinate systems and drawing modes.
- RGB colors.
- Making things move in straight lines.
- Hit detection for rectangles and circles.
- Special effects, such as gravity, fading, throwing, easing, and drag-and-drop.
- Strings and fonts.
- Sound in Processing using the Minim library.
- Rotations and translations.
- Arrays and loops.
- Creating a vector helper class.
- Creating a sprite class for animated objects.
- Combining it all into games.

Grading:
Programming assignments will be given throughout the course, along with a midterm exam and final exam. A more detailed marking scheme will be given in the first week of classes.

Students must attain an overall passing grade on the weighted average of exams in the course in order to obtain a clear pass (C- or better).

Reference Books:


Processing: Creative Coding and Computational Art, Ira Greenberg, Friends of Ed, 2007, 9781590596173, Lots of good examples of interesting and useful programming techniques.


Academic Honesty Statement:
Academic honesty plays a key role in our efforts to maintain a high standard of academic excellence and integrity. Students are advised that ALL acts of intellectual dishonesty will be handled in accordance with the SFU Academic Honesty and Student Conduct Policies (http://www.sfu.ca/policies/gazette/student.html).