

# COMPUTER SCIENCE PROGRAM YOUR OWN GAME

## Preliminary Course Syllabus

Iowa State University  
Brian Nakayama  
April 27<sup>th</sup>, 2014

From day 1, students will be evenly divided amongst teams. Each day will include interactive activities, where students will have chances to win points for their team by answering questions correctly. At the end of the class, a prize will be awarded randomly to a team according to a probability equal to their points divided by the points of the class.

- Rationale: Choosing the prize by a probability gives all students a chance to win the prize, keeping students with less points motivated.

Students will be selected to answer questions based on how many questions they've answered correctly.

- Goal: All students should answer approximately the same number of questions correctly each day.

At the end of each day a paper work sheet will be handed out for homework.

- Rationale: Sitting in front of a computer for more than 6 hours a day would be too demanding (plus bad for morale, etc.).

The worksheets primarily are for introducing students to new topics we will cover the next day or for reviewing essential topics. Grading of the worksheets is best accomplished during lunch, so that the homework can be reviewed when the students come back.

Below is a tentative schedule of what will be done each day. It has been revised with lessons learned from the class taught in 2014.

### Day 1:

- 20 minutes - questionnaire on name, why they're interested, and programming experience.
- 40 minutes - explain what the class is, what Java is and what the class shall teach.
- 30 minutes - do Hello World and compile using the terminal.
- 30 minutes - start slightly harder Hello World example that takes argument from the terminal.
- 1 hour short lecture on different types of variables, what they represent, and how they are used.
- --break--
- 1 hour - continue working on the extended Hello World program. Introduce "if" statements. Review print and println methods and special characters.
- 1 hour - introduce scanner class, as well as basic arithmetic for numbers and different number types. Start a basic text based RPG to practice basic understanding.
- 1 hour - introduce while loops.

### Day 2:

- Depending on the student's level of understanding student may be given different advanced tasks that will help students learn about a different area in Computer Science while doing similar programming activities with the class.
- 20 minutes - go over homework.
- 40 minutes - introduce using Eclipse
- 1 hour - introduce arrays, for loops, and switch-case statements. Do quizGame example.

- 1 hour - continue to develop RPG by giving it a map.
- --break--
- 1 hour - lecture on objects (static vs. nonstatic), methods and references.
- 1 hour - practice using methods, and also refactor RPG into methods.
- 30 minute - lecture on recursion.
- 30 minute - factorial example.

#### Day 3:

- 1 hour – review objects and methods.
- 1 hour - introduce File I/O, and exception handling.
- 1 hour - practice File I/O and diary example.
- --break--
- 20 minutes - go over homework.
- 1 hour - implement File I/O to save/open maps and games on RPG.
- 40 minutes - lecture on scope blocks of codes, and an activity.
- 1 hour - lecture on scope of variables: private, protected, default, and public.

#### Day 4:

- 1 hour - review scope
- 1 hour - introduce graphics and the game engine
- 1 hour - interactive activity to get projects started
- --break--
- 20 minutes - go over homework
- 1 hour - lecture on other programming languages and their uses. Students may work on concept for their game.
- 1 hour - work time.
- 1 hour - lecture on binary numbers.

#### Day 5:

- 1 hour - abstract classes and interfaces
- 30 minutes - introduction to bitwise operators and ternary operators.
- 30 minutes - in class exercise.
- 1 hour - continue working on RPG.
- --break--
- 20 minutes - go over homework.
- 40 minutes - lecture on historical Computer Scientists (Mothers/Fathers of Computer Science).
- 1 hour - work time.
- 30 minutes - lecture on how to create an executable jar so that they can show their final project to friends and family.
- 30 minutes - work time and copy student's work for writing letters to parents.

#### Day 6:

- Help students set up their demo.