Course Outcomes Guide 2013

Course Title: CSC/IST 134 Introduction to Java Programming

Course Instructor(s): Doug Leisher

Programs: Computer Science, Information Systems Technology - Developer

Expected Learning Outcomes

1. Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions
2. Apply the techniques of structured (functional) decomposition to break a program into smaller pieces.
3. Create program code in the traditional programming style using a non-graphical user interface and in modern style programming using a graphical user interface
4. Discuss topics pertinent to today's computer programming environment

Assessment (How do students demonstrate achievement of these outcomes?)

1. Successful compilation and execution of program code
2. Satisfactory scores on exams and programs
3. Projects are graded on a rubric that meets the professional standards of developing code
4. Students give a presentation pertaining to computer programming

Validation (What methods are used to validate your assessment?)

1. Faculty Review
2. Approval of Computer Science Department of a 4 year transfer university
3. Validation from IST Advisory Committee (comprised of area business people who hire our graduates).
4. Students complete a survey regarding various aspects of the course and instructor
5. Since computer programming is cumulative, program assignments at the end of the course incorporate most, if not all, topics taught in the course. For example, early in the course students learn the importance and use of variables. Variables are then used in every program a student writes from that point forward.

Results (What do the data show?)
Students enjoyed creating programs using Java. They especially liked embedding Java programs onto web pages. The presentation was also received very well. Students liked hearing material presented by other students, especially topics such as programming an app for a smartphone.

Follow-up (How have you used the data to improve student learning?)
Content coverage was discussion by the students. The instructor did not start the class. The students were responsible for discussing the information and helping their classmates that did not understand. The instructor interjected topics that should be covered only if the students did not bring them up. Additional textbooks were brought to class to use as additional research. An excellent website (java.sun.com) was found by a student as a resource and tutorial.

Also, students seem to like being introduced to topics then allowed to use much class time to work on assignments.

Programming is a course whereby students learn much more by trying and debugging programs (asking for help when needed) than by having an instructor tell them exactly what to do.

At the end of each semester, the instructor analyzes student results from the last couple semesters so as to compare results. This helps define trends so that instruction techniques/assignments can be adjusted to better help future students learn the material.

Budget Justification

(What resources are necessary to improve student learning?)

- At this point, the software is a free download so the Java development environment just needs installed on computer lab PCs where this is to be taught
- Use of a computer lab that has a projector for instructor to project screen to students
- Access to computers with Java other than during class time
- It would be nice to be able to bring in a speaker who actually uses Java in their business