

Minimal competencies for CSC 221 – Approved 4/14/00

Introduction to Computer Science II

Objects

- The student shall have successfully developed applications programs that exploit programmer-defined objects. In particular, the student shall create at least two applications programs that each utilizes at least four separately compilable class definition files, which were created by the student.

Text and Binary File I/O

- The student shall have successfully written applications modules that are capable of reading from and writing to data files using low-level, high-level, and object-level binary file I/O. In addition, each student shall develop applications modules to read from and write to text files using ASCII file I/O.

Sorting

- The student shall have successfully written and demonstrated applications modules for sorting using at least one $O(n^2)$ method and one $O(n \log n)$ method.

Searching

- The student shall have successfully written and demonstrated applications modules for searching an array linearly and logarithmically.

GUIs

- The student shall have successfully written and demonstrated applications modules that utilize Java event handling to create menus, buttons, text fields, and label fields, as well as to monitor and use mouse events. In addition, the student shall have successfully implemented code to use events and listeners.
- The student shall have successfully written and demonstrated applets that utilize Java graphics. In particular, the applet(s) must:
 - create and reveal a graphical display window;
 - draw lines, circles, and rectangles with arbitrary dimensions and locations;
 - change drawing colors and clear the drawing area.

Recursion

- The student shall develop at least one application module that employs recursion.

Inheritance and polymorphism

- The student shall have successfully written and demonstrated applications modules that utilize inheritance, polymorphism, and code reuse in at least two programs.
- The student shall have successfully written and demonstrated application classes that utilize abstract classes and an introduction to interfaces.

Software Development Teamwork

- The student shall actively participate on a software development team to create a software product that demonstrates collaborative capability.