CS 116 Introduction to Object-Oriented Programming II

Instructor: Boris Glavic, Email: `bglavic@iit.edu`

Instructor Webpage: `www.cs.iit.edu/~glavic`

Course Webpage: `www.cs.iit.edu/~glavic/cs116`

Course Description

Continuation of CS 115. Introduces more advanced elements of object-oriented programming – including dynamic data structures, recursion, searching and sorting, and advanced object-oriented programming techniques. For students in CS and CS related degree programs.

2 credit hours; required for CS & CPE (or CS201); 75 min. lecture & 75 min. lab each week

Course Material

The following text book is required reading material for the course.

Julie Anderson and Hervé J. Franceschi, **Java Illuminated**, Jones & Bartlett Learning, **ISBN-13:** 978-1284140996

Other good more advanced Java books are:

Joshua Bloch, **Effective Java**, Addison Wesley, **ISBN-13:** 978-0134685991

Notebooks, labs, and slides are made available through the course webpage

Prerequisites:

- *Courses:* CS 115 (2-1-2)

Students with Disabilities

Reasonable accommodations will be made for students with documented disabilities. In order to receive accommodations, students must obtain a letter of accommodation from the Center for Disability Resources. The Center for Disability Resources (CDR) is located in 3424 S. State St., room 1C3-2 (on the first floor), telephone 312 567.5744 or `disabilities@iit.edu`.

Piazza

his term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com.

The link to the Piazza webpage can be found here: `http://cs.iit.edu/~glavic/cs116`

Course Objectives

Students should be able to:

- Analyze and explain the behavior of simple programs involving the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, functions, pointers

- Write a program that uses each of the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, functions, pointers

- Break a problem into logical pieces that can be solved (programmed) independently.

- Develop, and analyze, algorithms for solving simple problems.

- Use a suitable programming language, and development environment, to implement, test, and debug algorithms for solving simple problems.

- Write programs that use each of the following data structures (and describe how they are represented in memory): strings, arrays, structures, and class libraries including strings and vectors

- Explain the basics of the concept of recursion.

- Write, test, and debug simple recursive functions and procedures.

- Explain and apply object-oriented design and testing involving the following concepts: data abstraction, encapsulation, information hiding, sub-classing, inheritance, templates

- Use a development environment to design, code, test, and debug simple programs, including multi-file source projects, in an object-oriented programming language.

- Solve problems by creating and using sequential search, binary search, and quadratic sorting algorithms (selection, insertion)

- Determine the time complexity of simple algorithms.

Course Topics
The following topics will be covered in the course:

- The Java language

  - Java's type system and syntax
  - The Java class library
  - Documentation
  - Exception Handling

- Object-oriented Programming and Design

  - Classes, Instances, Inheritance, and Polymorphism
  - Design Patterns

- Debugging, Logging, and Testing

- Data Structures

  - Lists
  - Maps
  - Trees

- Basic Algorithms

  - Searching and Sorting
  - Computational Complexity and Runtime Analysis

- Programming Techniques

  - Recursion
  - Concurrent Programming

- Tool ecosystem

  - Editors and IDEs
  - Java binaries
  - Version Control

Workload and Grading Policies:

Course Project:
There will be a programming project in the later part of the course.

Midterm and Final Exam:
There will be a midterm and final exam covering the topics of the course.

Lab Assignments:
There will be several lab assignments during the course. The main objective of these assignments is for students to employ the concepts and language features learned in class in a practical context.

Grading Policies:

See the course webpage for policies regarding late lab assignments and plagiarism.

- Lab assignments: 30%

- Project: 10%

- Midterm Exam: 25%

- Final Exam: 35%

The grading standard is:

- 90+ = A

- 80+ = B

- 70+ = C

- 60+ = D

- <60 = E