Course Code     Course Title
**COMP 1022P     Introduction to Computing with Java**

Course Description

This course is designed to equip students with the fundamental concepts of programming elements and data abstraction using Java. Students will learn how to write procedural programs using variables, arrays, control statements, loops, recursion, data abstraction and objects using an integrated development environment. Exclusion(s): COMP 1021, COMP 1022Q (prior to 2020-21), COMP 2011, COMP2012H, ISOM 3320; Mode of Delivery: [ONL] Pure online delivery

List of Topics

1. Course Logistics
2. Problem Solving, Introduction to Computer Programming
3. Program Comments, Primitive Data Types, Variables and Literals, Console I/O
4. Arithmetic Operations, Assignment Operations, Relational Operations, Logical Operations, Precedence of Operators, Manual Type Casting
5. Checking and Selection Statements
6. Looping Statements
7. Arrays
8. Classes and Objects
9. Inheritance
10. Static Variables and Methods
11. Recursion
12. File I/O
13. Strings
14. Abstract Data Types (Stack and Queue)

This course is comprised of two 5-module parts.

Part 1 introduces programming fundamentals.

- Problem solving
- Primitive data types and arithmetic expressions
- Object-oriented programming basics
- Branching and Loops
- Arrays

Part 2 covers the following topics.

- String manipulation
- File I/O

- Simple event-driven programming
- Recursion
- Abstract data types

Textbook

Introduction to Java Programming and Data Structures: Comprehensive Version, Liang, Y. Daniel, Pearson, c2020, 12th Edition.
- Library copy (10th Edition): QA76.73.J38 L533 2015
- Number of pages: 1320
- ISBN: 9780133761313

Reference book

Java How To Program (Early Objects), Harvey Deitel & Paul Deitel, Pearson Education, c2017, 11th Edition.
- Library copy (9th Edition): QA76.73.J38 D45 2012
- Number of pages: 1496
- ISBN: 9780132575669

Grading Scheme

| Online activities | 10% |
|---|---|
| Labs | 10% |
| Project | 10% |
| Midterm examination | 30% |
| Final examination | 40% |
| Total | 100% |

Course Intended Learning Outcomes

On successful completion of this course, students are expected to be able to:
1. Demonstrate programming skills, with an emphasis on the Java programming language.
2. Design and develop effective solutions to computational problem using Java as a programming tool.

Assess Rubrics

| Level of Achievement | CILO 1 Demonstrate programming skills, with an emphasis on the Java programming language. | CILO 2 Design and develop effective solutions to computational problem using Java as a programming tool. |
|---|---|---|
| Exemplary (A- to A+) | The student is able to proficiently program in the Java language with no or very few errors, and is able to detect and fix bugs in the code on his/her own. The program can return the correct output on all input instances. | Given a problem definition, the student is able to proficiently write a Java program to solve the problem. |
| Competent (C to B+) | The student is able to program in the Java language satisfactorily. The code may contain a few bugs, but the student is able to identify the majority of the bugs on his/her own. Given some further help, the student is able to identify and fix all the bugs. The program can return the correct output on most input instances, but may miss some special or boundary cases. | Given a problem definition, the student is able to write a Java program that solves the problem satisfactorily. The program may crash or behave incorrectly under some harsh tests. |
| Needs Work (D to C-) | The student is able to barely program in the Java language on his/her own, but with some help, he/she is able to write a program that behaves correctly on typical input instances. The code often contains bugs and the student is hardly able to identify or correct them. | Given a problem definition, the student is barely able to write a Java program that solves the problem. The program often crashes or behaves incorrectly. |
| Unsatisfactory (F) | The student is not able to program in the Java language. | Given a problem definition, the student is not able to use Java to solve it. |