

Course Syllabus

Description:

Develop the skills required to write programs or parts of programs to correctly solve specific problems. You will learn design techniques to make programs understandable, adaptable, and reusable. This course provides elective credit only.

Estimated Completion Time: 2 segments/ 32-36 weeks

Major Topics and Concepts:

Segment I:

Module 01

- 01.00 Orientation
- 01.01 Course Folder Management
- 01.02 Installing Java
- 01.03 Installing the BlueJ IDE
- 01.04 BlueJ Tutorial
- 01.05 Hello World
- 01.06 Stylish Java
- 01.07 Checkpoint Alpha

Module 02

- 02.00 Introduction
- 02.01 Order of Operations
- 02.02 Printing Arithmetic Expressions
- 02.03 Primitive Data Types: ints
- 02.04 Primitive Data Types: doubles
- 02.05 Arithmetic Expressions
- 02.06 Primitive Data Type Conversions
- 02.07 Pitfalls, Surprises, and Shortcuts
- 02.08 Challenge Program

Module 03

- 03.00 Introduction
- 03.01 Pseudocode, Recipe for Success
- 03.02 Primitive Data Types: char
- 03.03 String Objects Lite
- 03.04 Escape Sequences
- 03.05 The Java API
- 03.06 String Class Methods: The Basics
- 03.07 Scanner Class Methods
- 03.08 Parsing
- 03.09 Challenge Program
- 03.10 Discussion-Based Assessment
- 03.11 Module 3 Exam
- 03.12 Checkpoint Gamma

Module 04

- 04.00 Getting Started with if Statements
- 04.01 Number Systems
- 04.02 Primitive Data Types: booleans
- 04.03 Condition Statements: if
- 04.04 Condition Statements: if-else
- 04.05 Condition Statements: if-else-if
- 04.06 Comparing Strings
- 04.07 Logical Operators
- 04.08 Checkpoint Delta

Module 05

- 05.00 Getting Started with loops
- 05.01 while Loops (Part 1)
- 05.02 while Loops (Part 2)
- 05.03 Reading Text Files
- 05.04 for Loops
- 05.05 Nested Loops
- 05.06 Writing Text Files
- 05.07 Challenge Program
- 05.08 Checkpoint Epsilon
- 05.09 Discussions

Module 06

- 06.00 Getting Started with Arrays
- 06.01 One Dimensional Arrays
- 06.02 Formatting Output
- 06.03 for-each Loops
- 06.04 Challenge Program
- 06.05 Challenge Exam Part 1
- 06.05 Challenge Exam Part 2

Module 07

- 07.00 Getting Started with Methods
- 07.01 Java's Math Class
- 07.02 Defining New Static Methods: Part 1
- 07.03 Defining New Static Methods: Part 2
- 07.04 Defining New Static Methods: Part 3
- 07.05 Discussion-Based Assessment
- 07.06 Challenge Program
- 07.07 Checkpoint

Module 08

- 08.00 Getting Started with Objects
- 08.01 Real World Objects
- 08.02 Instances of a Class
- 08.03 Default Constructors
- 08.04 Discussion Topic
- 08.05 Constructors with Parameters

- 08.05 Constructors with Parameters
- 08.06 Overloading Methods and Using Two Classes
- 08.07 Constructing Multiple Objects
- 08.08 Arrays of Objects
- 08.09 Java Docs
- 08.10 ArrayLists I
- 08.11 ArrayLists II
- 08.12 Challenge Program
- 08.13 Checkpoint Theta
- 08.14 Challenge Exam Part 1
- 08.14 Challenge Exam Part 2

Module 09

- 09.00 What is a Computer?
- 09.01 Computer Anatomy 101
- 09.02 Computer History: Back in the Day
- 09.03 Four Generations of Modern Computers
- 09.04 Challenge Program
- 09.05 Checkpoint Iota

Module 10

- 10.00 Semester Exam

Segment II

Module 11

- 11.00 Technology and Society
- 11.01 Privacy Issues
- 11.02 Security Issues
- 11.03 Legal Issues
- 11.04 Future Issues
- 11.05 Computer Science

Module 12

- 12.00 Getting Started with Recursion
- 12.01 Divide et Impera
- 12.02 Real World Recursion
- 12.03 The Recursive Leap of Faith
- 12.04 There and Back Again
- 12.05 Are We There Yet?
- 12.06 Challenge Program
- 12.07 Mystery Message
- 12.08 Create Your Own Challenge Exam
- 12.09 Computer Science Lab: Magpie
- 12.10 Discussion-Based Assessment
- 12.11 Check Point Lamda

Module 13

- 13.00 Introduction to Inheritance and Polymorphism

- 13.01 Extending Classes
- 13.02 Class Hierarchies
- 13.03 Polymorphism
- 13.04 Overriding Methods
- 13.05 Computer Science Lab: Magpie
- 13.06 Checklist

Module 14

- 14.00 Getting Back to Basics
- 14.01 Design Strategy: Iterative and Incremental
- 14.02 Static Means Never Having to Instantiate an Object
- 14.03 Class Variables and Constants
- 14.04 Revisiting Randomness
- 14.05 this or That Variable
- 14.06 Thinking Outside the Box
- 14.07 Challenge Program
- 14.08 Checkpoint
- 14.09 Challenge Exam
- 14.10 Discussion
- 14.11 Computer Science Lab: Picture

Module 15

- 15.00 Introduction to Abstractions
- 15.01 Abstract Classes
- 15.02 Built-In Interfaces
- 15.03 Interfaces
- 15.04 Comparable Interface
- 15.05 Challenge Program
- 15.06 Challenge Exam Part 1
- 15.06 Challenge Exam Part 2
- 15.07 Computer Science Lab: Picture
- 15.08 Checklist

Module 16

- 16.00 Introduction to Standard Algorithms
- 16.01 Traversals Lesson
- 16.02 Replacements Lesson
- 16.03 Insertions Lesson
- 16.04 Deletions Lesson
- 16.05 Challenge Program
- 16.06 Computer Science Lab: Picture
- 16.07 Checklist

Module 17

- 17.00 Introduction to Sorting
- 17.01 Bubble Sort
- 17.02 Insertion Sort
- 17.03 Selection Sort
- 17.04 Merge Sort
- 17.05 Challenge Program

- 17.05 Challenge Program
- 17.06 Computer Science Lab: Elevens
- 17.07 Checklist

Module 18

- 18.00 Introduction to Searching
- 18.01 Sequential Search
- 18.02 Binary Search
- 18.03 Challenge Program
- 18.04 Challenge Exam
- 18.05 Computer Science Lab: Elevens
- 18.06 Checklist
- 18.07 Discussion-Based Assessment

Module 19

- 19.00 Introduction to Program Analysis
- 19.01 Assertions and Exceptions
- 19.02 Challenge Program
- 19.03 Computer Science Lab: Elevens
- 19.04 Checklist

Module 20

- 20.00 Getting Started with Your Review
- 20.01 Exam Format, Grading, Hints
- 20.02 Java Features, Part 1
- 20.03 Java Features, Part 2
- 20.04 Program Design and OOP Concepts
- 20.05 Algorithms
- 20.06 Solutions to Past Free Response Questions
- 20.07 Practice Exams
- 20.08 Reflections
- 20.09 Final Exam

Course Assessment and Participation Requirements:

To achieve success, students are expected to submit work in each course weekly. Students can learn at their own pace; however, “any pace” still means that students must make progress in the course every week. To measure learning, students complete self-checks, practice lessons, multiple choice questions, projects, discussion-based assessments, and discussions. Students are expected to maintain regular contact with teachers; the minimum requirement is monthly. When teachers, students, and parents work together, students are successful.