

Revised Fall 2016

# ITP120

## Java Programming I

### COURSE OUTLINE

#### Prerequisites:

ITP100 of instructor's permission

#### Course Description:

**Entails instruction in fundamentals of object-oriented programming using Java. Emphasizes program construction, algorithm development, coding, debugging, and documentation of console and graphical user interface applications.**

ITP 120 is the first semester of Java programming. The Java programming language is one of the most important computer languages for both client-side and server-side applications. Java allows you write intranet applications and other e-business solutions that are the foundation of corporate computing. It also is integral to the development of client-side applications for everything from games to Droid applications.

This course introduces the fundamentals of general computer programming and specifically, Java. Topics include a review of constructs from ITP100 including decisions, loops, and method calls. Collections are introduced and include arrays and ArrayLists.

A major emphasis of the course concentrates on Object Oriented programming, and the concepts of inheritance, encapsulation, and polymorphism which are foundations for successful Java programming.

Flat file and serializable file input and output are discussed. A wrap up project concentrates on a menu driven application that encapsulates the major topics of the class.

**Semester Credits: 4 Lecture Hours: 4 Lab/Recitation Hours: 0**

VIRGINIA WESTERN COMMUNITY COLLEGE  
PO Box 14007  
Roanoke, VA 24038  
(540)-857-7273



# ITP120 Java Programming I

## Course Outcomes

**At the completion of this course, the student should be able to:**

- Have an introduction to computers, programs, and Java
- Understanding the fundamentals of input, processing, output
- Understand how to use both console and dialog box input and output
- Be able to master the concepts of Java constructs including looping and selections
- Understand the concept of static entities
- Be able to modularize code with methods
- Understand how to utilize one dimensional arrays
- Understand how to apply Java objects and classes to solve programs
- Be able to write code with classes and associations
- Understand and utilize the concepts of inheritance and polymorphism
- Understand how to use flat files
- Be able to write code that catches, handles, and throws exceptions
- Understand the concepts of abstract classes
- Be able to implement interfaces to help with generic solutions
- Be able to utilize ArrayLists for collections



# ITP120 Java Programming I

## Required Materials:

Eclipse Software (Open source supplied by the instructor)

## Textbook:

**Java, How to Program, 10<sup>th</sup> edition**, Paul and Harvey Deitel, ISBN 978-0133807806 Note:  
There is also a Kindle edition and I will supply a free version you can read online through Safari  
Online

The following supplementary materials are available:

1. Safari Online is available for students to view other Java books online for free
2. Internet searches are a fundamental source for auxiliary materials on Java
3. Materials from the Oracle Academy of which Virginia Western is a member
4. Video movies provided by the instructor.



# ITP120 Java Programming I

Topical Description: (Outline chapters and sections to be covered in the book – may include timeline)

Daily or Weekly Schedule (NOTE: Modules are two weeks in length):  
**EACH MODULE REQUIRES A LABORATORY AND QUIZ TO BE SUBMITTED IN BLACKBOARD**

## **Module 1:** Introduction to Java, Classes, and Objects

Learning objectives:

1. Understand how to declare a Java class and a the main method
2. Understand input and output to Java programs
3. Be able to run Java programs in Eclipse
4. Be able to import and save Java applications as .jar files
5. Understand Java arithmetic
6. Be familiar with equality and relational operators
7. Know the common primitive data types
8. Understand Java expression and how to write them

Reading Assignment:

1. Textbook – Chapters 1-2 (just skim chapter 1)
2. CodeConventions.pdf (included in the assignment files)

Laboratory Assignment:

1. Laboratory 1

## **Module 2:** Objects, classes, and decisions

Learning objectives:

1. Understand the concept of objects and classes
2. Be able to declare and use simple Java classes
3. Understand getters and setters
4. Understand the difference between primitive and reference data types
5. Understand program flow
6. Understand how to program if and if/else statements
7. Understand how to use Boolean expressions and how to compare data
8. Understand the do and the while construct



Reading Assignment:

1. Textbook –Chapters 3 (skip 3.8) and 4 (to section 4.6)

Laboratory Assignment:

1. Laboratory 2

**Module 3:** More constructs, loops

Learning objectives:

1. Be able to use the for construct
2. Understand what an iterator is and how to use one
3. Understand the logical operators
4. More on classes
5. Begin writing your own Java classes

Reading Assignment:

2. Textbook –Chapter 4 (4.7-4.13) and Chapter 5 (skip 5.10)

Laboratory Assignment:

2. Laboratory 3

**Module 4:** More Classes and Object Oriented Design. Lots of methods

Learning objectives:

1. Methods, methods and more methods!!!
2. Be able to design more complete Java class
3. Understand what constructors are and how to use them
4. Understand the relationship between Java objects
5. Understand method overloading
6. Understand constructor overloading
7. Be able to describe the different access modifiers
8. Understand **has-a** and **is-a**
9. Understand the concept of **static**
10. Understand how to read APIs
11. Understand how to use APIs
12. Be able to use methods in the String, StringBuilder, and Character classes

Reading Assignments:

1. Textbook – Chapters 6 and 8 (skip 6.13, 8.16), Also 16.1-16.5

Laboratory Assignment



1. Laboratory 4

**Module 5:** Arrays, ArrayLists, and other Collections

Learning objectives:

1. Understand how to use arrays in Java
2. Understand how to declare and create arrays in Java
3. Understand the enhanced **for** statement
4. Be able to pass arrays into methods
5. Understand the Arrays class
6. Understand alternatives (ArrayLists, etc)
7. Become familiar with the Java Collections API

Reading Assignment:

1. Textbook – Chapter 7 (skip 7.9-7.10, 7.15)

Laboratory Assignment

1. Laboratory 5

**Module 6:** Advanced Inheritance Topics and Interfaces

Learning Objectives:

1. Understand class hierarchies and the ramifications
2. Understand the relationships between super classes and subclasses
3. Understand method overriding
4. Be able to design for inheritance
5. Understand the **Object** class
6. Be able to discuss polymorphic behavior
7. Understand how to use **abstract** classes
8. Understand the concept of interfaces and their use
9. Be familiar with common searching and sorting algorithms
10. Understand designing for interfaces

Reading Assignment

1. Chapters 9 and 10 (skip 9.8, 10.8)

Laboratory Assignment

1. Laboratory 6

**Module 7:** I/O, Exceptions, and JavaDocs

Learning Objectives:

1. Understand Java Exceptions Hierarchy



2. Understand how to use a try/catch/finally structure
3. Understand how to read and write text files
4. Understand how to read and write object oriented serialized files
5. Understand how to use the Java **File** and **JFileChooser** classes
6. Understand exception handling
7. Understand how to create JavaDocs

Reading Assignment

1. Chapter 11 and 17
2. Appendix M

Laboratory Assignment

1. Laboratory 7

**Module 8:** Generic Collections

Learning Objectives:

1. Collections Overview
2. Wrapper classes
3. Auto-boxing and Auto-unboxing
4. Interface Collection and Class Collections
5. Lists
6. Collections Methods
7. Sets
8. Maps
9. Properties Class
10. Synchronized and Unmodifiable Collections
11. Abstract Implementations

Reading Assignment

1. Textbook –Chapter 20

Laboratory Assignment

1. Laboratory 8



# ITP120 Java Programming I

## Notes to Instructors

(List information about optional topics, departmental exams, etc)

1. Each module is two weeks long and requires both a quiz and a laboratory assignment submitted through Blackboard
2. A midterm exam is required.
3. A semester long project is required

