

Programming II Java: Grades 10, 11, 12

Adopted 2008

Review Programming Techniques, Ethics, and Privacy

1.1 Discuss the ethical and privacy issues of programming

1. Identify ethical and privacy practices in computer programming [1.1.1](#)
-

1.2 List the steps of the programming process

1. Identify correct step when given an example [1.2.1](#)
-

Data Validation

2.1 Define terminology

1. Manipulate vocabulary and concepts individually and in groups [2.1.1](#)
-

2.2 Explain the importance of data validation

1. Give examples of good data validation rules for a variety of situations [2.2.1](#)
-

2.3 Explain the logic of numeric range checks

1. Write programs that use range checks [2.3.1](#)
-

2.4 Explain the logic of data validation to match a particular pattern

1. Write programs that require data to fit a specified pattern (i.e., Social Security Number 123-45-6789, phone number 123-456-7890, etc.) [2.4.1](#)
-

String Manipulation

3.1 Define terminology

1. Manipulate vocabulary and concepts individually and in groups [3.1.1](#)
-

3.2 Explain the syntax and features of various commands dealing with ASCII or Unicode numbers and their corresponding characters

1. Write program lines to determine the ASCII number of a character [3.2.1](#)
2. Write programs to use the ASCII number to print the corresponding character [3.2.2](#)

3.3 Explain the syntax and purpose of commands that handle all or part of a string and that concatenate strings

1. Write programs to determine the number of characters in a string [3.3.1](#)
 2. Write programs to print a particular group of characters that are contained in a string [3.3.2](#)
 3. Write programs that concatenate multiple strings into one [3.3.3](#)
-

3.4 Explain the reasons why breaking a string into its component parts is important

1. Write code that will take the first part of a string from a longer string (i.e., taking the area code from a telephone number) [3.4.1](#)
 2. Write code that will take characters from the middle of a string (i.e., removing the middle name from the full name) [3.4.2](#)
 3. Write code that will take characters from the right side of a string (i.e., ZIP code from the address) [3.4.3](#)
-

Classes and Objects

4.1 Define terminology

1. Manipulate vocabulary and concepts individually and in groups [4.1.1](#)
-

4.2 Explain the advantages of using classes

1. Define real-world entities in terms of state and behavior. [4.2.1](#)
-

4.3 Explain the format of simple classes

1. Write a class that contains data members and methods [4.3.1](#)
 2. Write a driver class that instantiates an object [4.3.2](#)
-

4.4 Explain the visibility modifiers public and private

1. Write a class that has private data members and public methods [4.4.1](#)
-

4.5 Explain constructor methods

1. Write a class that contains a constructor with no arguments [4.5.1](#)
 2. Write a class that contains a constructor with one or more arguments [4.5.2](#)
 3. Write a class that overloads the constructor [4.5.3](#)
-

Methods

5.1 Define terminology

1. Manipulate vocabulary and concepts individually and in groups [5.1.1](#)
-

5.2 Explain the difference between formal and actual parameters

1. Give examples of formal and actual parameters [5.2.1](#)

5.3 Explain the matching of actual parameters to formal parameters in the method call

1. Write programs that use parameters in method calls [5.3.1](#)
-

5.4 Explain the difference between passing primitive data types as parameters and passing objects as parameters

1. Write methods passing both primitive data types and String objects [5.4.1](#)
-

5.5 Differentiate between void and value returning methods

1. Write methods that return values (accessor or getter methods) [5.5.1](#)
 2. Write void methods (mutator or setter methods) [5.5.2](#)
-

5.6 Explain how to pass arrays, ArrayLists, and other objects as parameters

1. Write methods that have arrays, arraylists or other objects as parameters [5.6.1](#)
-

One-dimensional Arrays or Vectors**6.1 Define terminology**

1. Manipulate vocabulary and concepts individually and in groups [6.1.1](#)
-

6.2 Explain the logical steps in initializing and loading a one-dimensional array

1. Write an appropriate program to declare, instantiate and initialize a one-dimensional array [6.2.1](#)
-

6.3 Explain the use of subscripts and the syntax to use them

1. Use subscript to access particular elements in a one-dimensional array [6.3.1](#)
-

6.4 Explain the logical steps in traversing a one-dimensional array to perform calculations and comparisons

1. Write loops that traverse a one-dimensional array, performing calculations and comparisons [6.4.1](#)
-

6.5 Explain the logical steps in printing an entire array of data

1. Write loops that print the contents of a one-dimensional array [6.5.1](#)
-

6.6 Explain the logical steps to insert a value in a one-dimensional array

1. Write code that inserts values into an existing array [6.6.1](#)
-

6.7 Explain the logical steps in deleting elements in a one-dimensional array

1. Write code that deletes elements from an existing array [6.7.1](#)
-

6.8 Explain the use of parallel one-dimensional arrays or vectors

1. Write programs that contain parallel one-dimensional arrays or vectors [6.8.1](#)
-

ArrayLists, Searching and Sorting

7.1 Define terminology

1. Manipulate vocabulary and concepts individually and in groups [7.1.1](#)
-

7.2 Discuss the difference between an ArrayList and an array

1. Determine whether an array or ArrayList would be the best data structure for a given scenario [7.2.1](#)
 2. Discuss the differences between arrays and ArrayLists when adding and deleting [7.2.2](#)
 3. Discuss the differences between arrays and ArrayLists when accessing elements [7.2.3](#)
-

7.3 Know the difference between type specific and generic ArrayLists

1. Declare and instantiate both generic and type specific ArrayLists [7.3.1](#)
-

7.4 Know how to add, remove, and retrieve elements from an ArrayList

1. Write a program that adds, removes, and retrieves elements from an ArrayList [7.4.1](#)
-

7.5 Explain the difference between a binary search and a sequential search

1. Write programs to demonstrate binary and sequential searches [7.5.1](#)
-

7.6 Know the difference between insertion, selection, and merge sorts

1. When given code identify the type of sort used [7.6.1](#)