Practice Final

**You have 60 minutes to work on the exam. Answer each question to the best of your ability. If you are stumped, or don’t know how to write a specific piece of code, use comments, or pseudo code, to describe the process or steps necessary to write the program.**

1. Which of the following is valid for casting a String, “1234” to an integer?

a. (String) "1234";

b. 1234 + "";

c. Integer.parseInt("1234");

d. (int) "1234";

1. The main method for a class is written as:

a. public void static main(String[] args)

b. public static void main(String[] args)

c. public static void Main(String[] args)

d. public static void Main(string[] args)

1. Which of the following statements is true?

a. Every comment must end with a semicolon

b. Every statement in a program must end with a semicolon

c. Every method must end with a semicolon

d. Every line in a program must end with a semicolon

1. The extension name of a Java bytecode file is?

a. .java

b. .obj

c. .exe

d. .class

1. If a program compiles fine, but it produces incorrect result, then the program has what type of error?

a. compilation error b. runtime error c. logic error

1. Of the following, circle the answer(s) that will give a result of 0.5.

a. 1/2

b. 1.0/2

c. (double) (1 / 2)

d. (double) 1 / 2

1. The expression, ("Java " + 1 + 2 + 3), evaluates to which of the following:?

a. Java 123

b. Java123

c. Java6

d. Java 6

1. What is the output of the following code?

double x = 5.5;

int y = (int)x;

System.out.println("x is " + x + " and y is " + y);

a. x is 5.5 and y is 5.5

b. x is 5 and y is 5.5

c. x is 5.5 and y is 5

d. x is 5 and y is 5

1. What are the values of i and j after the following method is run?

public void add() {

int j = 1;

int i = ++j + j++ \* 5;

}

a. i = 17 and j = 3

b. i = 11 and j = 3

c. i = 16 and j = 3

d. i = 12 and j = 3

1. If you enter 1, 3, 2, 1, 3, into your keyboard when you run this program, what will be the output?

import java.util.Scanner;

public class Test1 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter five numbers: ");

double number3 = input.nextDouble();

input.nextDouble();

double number1 = input.nextDouble();

input.nextDouble();

double number2 = input.nextDouble();

// Compute average

double average = (number2 – number1 + number3 \* number1) / number2;

// Display result

System.out.println(average);

}

}

Output:­\_\_\_\_\_1.0\_\_\_\_\_\_

11. Write a static method named contains that accepts two arrays of integers *a1* and *a2* as parameters and that returns a boolean value indicating whether or not *a2*'s sequence of elements appears in *a1* (true for yes, false for no). The sequence of elements in *a2* may appear anywhere in *a1* but must appear consecutively and in the same order. For example, if variables called list1 and list2 store the following values:

int[] list1 = {1, 6, 2, 1, 4, 1, 2, 1, 8};

int[] list2 = {1, 2, 1};

Then the call of contains(list1, list2) should return true because list2's sequence of values {1, 2, 1} is contained in list1 starting at index 5. If list2 had stored the values {2, 1, 2}, the call of contains(list1, list2) would return false because list1 does not contain that sequence of values. Any two lists with identical elements are considered to contain each other, so a call such as contains(list1, list1) should return true.

You may assume that both arrays passed to your method will have lengths of at least 1. You may not use any Strings to help you solve this problem, nor methods that produce Strings such as Arrays.toString.

1 public class pattern {  
 2 public static void main(String[] args) {  
 3 int[] a1 = {1, 6, 2, 1, 4, 1, 2, 1, 8};  
 4 int[] a2 = {1, 0, 1};  
 5 // Is a2 contained in a1?  
 6 System.out.println(contains(a1, a2));  
 7 }  
 8   
 9 public static boolean contains(int[] a1, int[] a2) {  
10 // The current index of a2  
11 // Once we reach the ending index, we have a match  
12 int a2Index = 0;  
13   
14 // For each element in the first array, a1  
15 for(int i = 0; i < a1.length; i++) {  
16 // Go to the next number if we find a match  
17 if(a1[i] == a2[a2Index]) {  
18 a2Index++;  
19 } else {  
20 // Otherwise, if the number from a1[i]  
21 // does not match the number from a2[a2Index]  
22 // Start at the beginning of a2  
23 a2Index = 0;  
24 }  
25   
26 // If all of a2 has been traversed, then  
27 // we have successfully found the pattern   
28 if(a2Index == a2.length) {  
29 return true;  
30 }  
31 }  
32 // Return false after going through each number in a1  
33 return false;  
34 }  
35 }

12. Consider the following method. What are the values of the elements in array numbers[] after the following code executes?

public static void mystery3(int[] data, int x, int y) {

data[data[x]] = data[y];

data[y] = x;

}

int[] numbers = {3, 7, 1, 0, 25, 4, 18, -1, 5};

mystery3(numbers, 3, 1);

mystery3(numbers, 5, 6);

mystery3(numbers, 8, 4);

numbers[0]: \_\_\_\_\_\_7\_\_\_\_\_\_

numbers[1]: \_\_\_\_\_\_3\_\_\_\_\_\_

numbers[2]: \_\_\_\_\_\_1\_\_\_\_\_\_

numbers[3]: \_\_\_\_\_\_0\_\_\_\_\_\_

numbers[4]: \_\_\_\_\_\_8\_\_\_\_\_\_

numbers[5]: \_\_\_\_\_18\_\_\_\_\_\_

numbers[6]: \_\_\_\_\_\_5\_\_\_\_\_\_

numbers[7]: \_\_\_\_\_-1\_\_\_\_\_\_

numbers[8]: \_\_\_\_\_\_5\_\_\_\_\_\_

13. Write out the code for a Point class. Include all attributes, the constructor, the accessors and mutators, *setLocation(newX, newY)*, which sets the location of the point, a *slope(Point p2)* method, which returns the slope of two points as a double, and the toString method.

1 // This file defines a new class of objects named Point.  
 2 // This version of the class adds the following features:  
 3 // - Constructor: (Initializes new Point objects)  
 4 // - toString method: (Tells Java how to println Point objects)  
 5 // - Attributes: (Private fields to protect the object's data)  
 6 public class Point {  
 7 // Attributes: The properties that make up a Point  
 8 // Private means that only the Point object can access these variables  
 9 private int x; // Each Point object should have a variable  
10 private int y; // inside it named x, and a variable named y  
11   
12 // Initializes the state of a new Point object at the given x/y location.  
13 // 1) This method runs when new Point(x, y) is used in the main class file.  
14 // 2) Initializes the x and y coordinates of the object being created.  
15 public Point(int initialX, int initialY) {  
16 x = initialX;  
17 y = initialY;  
18 }  
19   
20 public static void main(String... args) {  
21 int j = 1;  
22 int i = ++j + j++ \* 5;  
23 System.out.println(i);  
24 System.out.println(j);  
25 }  
26   
27 // Getters  
28 // Returns the Point's x coordinate attribute.  
29 public int getX() {  
30 return x;  
31 }  
32 // Returns the Point's y coordinate attribute.  
33 public int getY() {  
34 return y;  
35 }  
36   
37   
38 // Setters  
39 public void setX(int newX) {  
40 x = newX;  
41 }  
42 public void setY(int newY) {  
43 y = newY;  
44 }  
45 // Sets the Point's x/y coordinates to a new location.  
46 public void setLocation(int newX, int newY) {  
47 x = newX;  
48 y = newY;  
49 }  
50   
51 // Actions  
52 // Computes the slope between this point and another Point, p2.  
53 public double slope(Point p2) {  
54 int x2 = p2.x;  
55 int y2 = p2.y;  
56 // Slope is defined as (y2 - y)/(x2 - x)  
57 double slope = (y2 - y)/(x2 - x);  
58 return slope;  
59 }  
60   
61 // This methods is used when System.out.println() is called  
62 public String toString() {  
63 // Return the x and y coordinates as a String: "(-5, 2)"  
64 return "(" + x + ", " + y + ")";  
65 }  
66 }  
67