

First Name: _____ Last Name: _____

McGill ID: _____ Section: _____

Faculty of Science
COMP-202A - Foundations of Computing (Fall 2013) - All Sections
Final Examination

December 5th, 2013
9:00 - 12:00

Examiners:

Jonathan Tremblay [Sections 3 (3:30-4:30) and 2 (1:00-2:30)]
Dan Pomerantz [Section 1 (10:30-11:30)]

Instructions:

• **DO NOT TURN THIS PAGE UNTIL INSTRUCTED**

- This is a **closed book** examination; only a letter-sized (8.5" by 11") **crib sheet** is permitted. This crib sheet can be single or double-sided; it can be handwritten or typed. Non-electronic translation dictionaries are permitted, but instructors and invigilators reserve the right to inspect them at any time during the examination.
- Besides the above, only writing implements (pens, pencils, erasers, pencil sharpeners, etc.) are allowed. The possession of any other tools or devices is prohibited.
- Answer **all** questions on the scantron sheet.
- This examination has **21** pages including this cover page, and is printed on both sides of the paper. On page 20, you will find information about **useful classes and methods**. **You may detach page 20 from the examination if you wish.**
- The Examination Security Monitor Program detects pairs of students with unusually similar answer patterns on multiple-choice exams. Data generated by this program can be used as admissible evidence, either to initiate or corroborate an investigation or a charge of cheating under Section 16 of the Code of Student Conduct and Disciplinary Procedures.
- **MAKE SURE TO WRITE YOUR NAME AND STUDENT ID ON THE SCANTRON AS WELL AS TO FILL IN THE BUBBLE PROPERLY AT THE BEGINNING OF THE EXAM**

Scoring

The exam will be scored as follows:

1. Questions 1 until 60 will be worth 1.5 points each (total of 90 points)
2. Questions 61 until 70 will be worth 1 point each (total of 10 points)

Regular multiple choice: 2 points each

1. What does the following program do:

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello world!");  
    }  
}
```

- (A) It does nothing.
- (B) It creates a pop up box with the text “Hello world!” in it that lets you click OK.
- (C) It prints a piece of paper with the text “Hello world!” on it.
- (D) It have the computer read “Hello world!” out loud.
- (E) It prints the text “Hello world!” to the screen.

2. Consider the following code. What prints in the following program?

```
public class Mystery {  
    public static int swap(int x, int y) {  
        int temp = y;  
        y = x;  
        x = temp;  
        return x;  
    }  
  
    public static void main(String[] args) {  
        int x = -3;  
        int y = -10;  
        swap(x, y);  
        System.out.println(x + " " + y);  
    }  
}
```

- (A) -3 -10
- (B) 0 0
- (C) -10 -3
- (D) -10 -10
- (E) -3 -3

3. Consider the following code. What prints in the following program?

```
public class Mystery {  
    public static void swap(int[] x, int[] y) {  
        int[] temp = y;  
        y = x;  
        x = temp;  
    }  
  
    public static void main(String[] args) {  
        int[] x = {1, 2};  
        int[] y = {3, 4};  
    }  
}
```

```
        swap(x, y);
        System.out.println(x[0] + " " + y[0]);
    }
}
```

- (A) 1 3
- (B) 1 2
- (C) 3 3
- (D) 2 3
- (E) 3 1

4. Consider the following code. What prints in the following program?

```
public class Mystery {
    public static void resize(int[] x) {
        x = new int[x.length * 2];
        System.out.print(x.length + " ");
    }

    public static void main(String[] args) {
        int[] x = {1, 2};
        resize(x);
        System.out.println(x.length);
    }
}
```

- (A) None of the above.
- (B) 4 2
- (C) 2 2
- (D) 2 4
- (E) 4 4

5. What is the type and value of the following expression?

`" " + (3 / 2) + (3 / 2)`

- (A) type is double, value is 2 (note: this is the same as 2.0)
- (B) type is String, value is 11 (note: this is the same as "11")
- (C) type is double, value is 3 (note: this is the same as 3.0)
- (D) type is String value is 1.51.5 (note: this is the same as "1.51.5")
- (E) type is int, value is 2

6. Suppose you have a method with the following header:

```
public static void fun(String x)
```

Which of the following are legal ways to call the method fun:

- I. `fun((String)3);`
- II. `fun(3 + "");`
- III. `fun("3");`

- (A) II and III
- (B) I, II, and III
- (C) I only
- (D) II only
- (E) III only

7. What prints in the following code?

```
int[] array = {3, 3, 2, 1, 3, 2, 1, 3, 3};
for (int i = 0; i < array.length; i+= 2) {
    if (array[i] == 3) {
        System.out.print("A");
    }
}
```

- (A) A
- (B) AAAAAA
- (C) AA
- (D) AAAA
- (E) AAA

8. Given char c, which of the following checks if the symbol stored in c is a numeric digit. (i.e. that printing the value of c would print a digit between 0 and 9)

- (A) ! (c < '0' && c > '9')
- (B) ! (c < 0 || c > 9)
- (C) c >= '0' && c <= '9'
- (D) c >= 0 && c <= 9
- (E) c <= 9

9. How many times does the letter B print in the following code snippet?

```
for (int i = 0; i < 10; i++) {
    System.out.println("B");

    for (int j = 0; j < 3; j++) {
        System.out.println("B");
    }

    System.out.println("B");
}
```

- (A) 30
- (B) 50
- (C) 45
- (D) 35
- (E) 40

10. Read the following code carefully. What prints in the following code?

```
int x = -1;
if (x < 0) {
    System.out.print("-");
}
if (x > 0) {
    System.out.print("+");
}
else {
    System.out.print("0");
}
```

- (A) -0
- (B) +-
- (C) -+0
- (D) -+
- (E) 0-

11. What is returned by the function `foo` if you input the value 6?

```
public static int foo(int n) {
    if (n == 0) {
        return 0;
    }

    return n * foo(n-1);
}
```

- (A) 720 (= $6 * 5 * 4 * 3 * 2 * 1$)
- (B) 120 (= $5 * 4 * 3 * 2 * 1$)
- (C) The program has an infinite recursion and never finishes.
- (D) 0
- (E) 1

12. What prints in the following code?

```
try {
    int[] a = new int[0];
    System.out.println(a[0]);
}
catch (NullPointerException n) {
    System.out.print("null ");
}
catch (ArrayIndexOutOfBoundsException a) {
    System.out.print("array out of bounds");
}
catch (Exception e) {
    System.out.print("error");
}
```

- (A) array out of bounds
- (B) The program crashes due to a run time exception.
- (C) null

- (D) array out of bounds error
- (E) null error

13. Which of the following types are stored as an address in Java?

- I. String[]
- II. Scanner
- III. String

- (A) II
- (B) I, II, III
- (C) I, III
- (D) I
- (E) I, II

14. Ted, Lilly, Barney, Robin, and Marshall are all friends. What is the biggest problem with the following code which is designed to store a list of Marshall's friends and Ted's friends?

```
// Create an array to store Marshall's Friends
String[] marshallsFriends = {"Ted", "Lilly", "Barney", "Robin"};

// Ted's friends are the same as Marshall's except that the
// name Marshall should replace Ted
String[] tedsFriends = marshallsFriends;
// Ted is not friends with himself so store Marshall into
// the first element of the array.
tedsFriends[0] = "Marshall";

// The goal is for marshallsFriends to store {"Ted", "Lilly", "Barney", "Robin"}
// and for tedsFriends to store {"Marshall", "Lilly", "Barney", "Robin"} at this point
```

- (A) The code has a bug which makes it so that Ted's first friend is himself.
- (B) The code has a bug which makes it so that Marshall's first friend is himself.
- (C) The code is inefficient because it takes 3 statements to do what we could do in 2 statements.
- (D) The code has a compiler error because you can't assign one array to another.
- (E) How I Met Your Mother isn't a good enough TV show to be the subject of a comp 202 exam question!

15. Given the knowledge that an ArrayList is implemented by storing a private property of an array with some unused elements at the end of the array, which of the following operations would you expect to be slowest on a large ArrayList?

- (A) getting the number of elements in the ArrayList
- (B) removing the last element of an already existing ArrayList
- (C) creating an empty ArrayList
- (D) removing the first element of an already existing ArrayList
- (E) getting the element at a specific index of the ArrayList

16. Suppose you have the following two loops:

```
for (int i = 0; i < n; i++) {  
    for (int j = 0; j < n; j++) {  
        System.out.println("HI!");  
    }  
}
```

Suppose that `n` is a variable of type `int` and when `n` equals 100 that the above code takes 2 seconds to execute. How long would you expect the code to take to execute if `n` equaled 200 and everything else stayed the same?

- (A) 2.01 seconds (i.e. a negligible amount longer than when `n` equals 100)
- (B) Infinite seconds
- (C) 6 seconds
- (D) 8 seconds
- (E) 4 seconds

17. Suppose you have the following two loops:

```
for (int i = 0; i < n / 2; i++) {  
    for (int j = 0; j < i; j++) {  
        System.out.println("HI!");  
    }  
}
```

Suppose that `n` is a variable of type `int` and when `n` equals 100 that the code takes 2 seconds to execute. How long would you expect it to execute if `n` equaled 200 and everything else stayed the same?

- (A) 6 seconds
- (B) 8 seconds
- (C) 2.01 seconds (i.e. a negligible amount longer than when `n` equals 100)
- (D) Infinite seconds
- (E) 4 seconds

18. Suppose you have the following two loops:

```
for (int i = 0; i < n / 2; i = i * 2) {  
    for (int j = 0; j < i; j = j * 2) {  
        System.out.println("HI!");  
    }  
}
```

Suppose that `n` is a variable of type `int` and when `n` equals 100 that the code takes 2 seconds to execute. How long would you expect it to execute if `n` equaled 200 and everything else stayed the same?

- (A) Infinite seconds
- (B) 6 seconds
- (C) 4 seconds
- (D) 2.01 seconds (i.e. a negligible amount longer than when `n` equals 100)
- (E) 8 seconds

19. Suppose we want to know the index where the `int` variable `g` is located in the following array.

```
int[] a = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

We use the following code,

```
for(int i = 0; i < a.length;i++)
{
    if(a[i] == g) {
        return i;
    }
}
```

Depending on the value of `g`, what is the maximum number of comparisons you would have to do in **the worst case**?

- (A) 1
- (B) 0
- (C) 18
- (D) 9
- (E) 4

20. Suppose we want to know the index where the `int` variable `g` is located in the following array, (same code as previous question)

```
int[] a = { 1,2,3,4,5,6,7,8,9 };
```

We use the following code,

```
for(int i = 0; i < a.length;i++)
{
    if(a[i] == g) {
        return i;
    }
}
```

What value of `g` causes the MOST comparisons?

- (A) 1
- (B) 2
- (C) -1
- (D) They are all the same.
- (E) 9

21. Suppose we want to know the index where the `int` variable `g` is in the following array, (same code as previous question)

```
int[] a = { 1,2,3,4,5,6,7,8,9 };
```

We use the following code,

```
for(int i = 0; i < a.length;i++)
{
    if(a[i] == g) {
        return i;
    }
}
```


What value of g causes the LEAST comparisons?

- (A) 9
- (B) 1
- (C) They are all the same.
- (D) -1
- (E) 2

22. Which line of code could be used to create an arraylist that would store integer values?

- (A) `int[] a1 = new int[n];`
- (B) `ArrayList<int> a1 = new ArrayList<int>();`
- (C) Impossible
- (D) `ArrayList a1 = new ArrayList();`
- (E) `ArrayList<Integer> a1 = new ArrayList<Integer>();`

23. Suppose `a` stores the address of a non-null array. What will writing the following do?

```
System.out.println(a[a.length]);
```

- (A) Print the first element of the array `a`
- (B) Print the last element of the array `a`
- (C) Cause an `ArrayIndexOutOfBoundsException`
- (D) None of the above.
- (E) Cause a `NullPointerException`

24. If you measured the size of the following variables in memory in Java, which would be largest?

- (A) `int[][] z`
- (B) It is impossible to say without knowing the length of the string and arrays.
- (C) `String x`
- (D) `int[] y`
- (E) They are all equal.

25. How many `ints` can be stored in the following multidimensional array.

```
int[][][] x = new int[2][3][2];
```

- (A) 12
- (B) 10
- (C) 11
- (D) 6
- (E) 7

Longer multiple choice questions (3 points each)

Object Oriented Programming

In the next several questions, you will start with a simple class and answer some questions about it. In the process, you will add to the class and then write another class that use this class.

Consider the following (incomplete) class used to represent a rectangular room in memory:

```
// This class is used to represent a room with rectangular dimensions.
public class RectangleRoom {
    // The name of the room. e.g. living room, kitchen, bathroom, etc
    public String name;

    private int width;

    private int height;

    private int length;
}
```

26. Suppose we wanted to write a method inside the `RectangleRoom` class that would allow another class access to the number of cubic feet of the room. (Note that you can calculate the cubic feet of a `RectangleRoom` by multiplying the width by the height by the length of a room).

What could the method header look like?

- (A) `private static int getSize(int width, int height, int length)`
 - (B) `public static int getSize(int width, int height, int length)`
 - (C) `public static int getSize()`
 - (D) `public int getSize()`
 - (E) `private int getSzie();`
27. Suppose we wanted to write a method that allowed one to simultaneously create a `RectangleRoom` object and initialize its values to any arbitrary values.

What could the method header look like?

- (A) `public void RectangleRoom(int width2, int height2, int length2, String name2)`
 - (B) `public RectangleRoom(int width2, int height2, int length2, String name2)`
 - (C) `public void RectangleRoom()`
 - (D) `public static RectangleRoom(int width2, int height2, int length2, String name2)`
 - (E) `public RectangleRoom RectangleRoom(int width2, int height2, int length2, String name2)`
28. Which of the following statements are true?
- I. Since `name` is `public` you can access it directly in another class
 - II. Since `name` is `public` you can not access it directly from within the class `RectangleRoom`
 - III. Since `width` is `private` you can not access it directly from another class.

- (A) II
- (B) All are true
- (C) I, II
- (D) I, III

- (E) I
29. Now suppose you want to define a new class `MontrealApartment` that uses the `RectangleRoom` class. Every `MontrealApartment` consists of several `RectangleRooms` and a price. To declare that every `Montreal` apartment should consist of an array of `RectangleRooms`, what would one write immediately after declaring the class?
- (A) `public static RectangleRoom[] rooms;`
 - (B) `public static RectangleRoom[];`
 - (C) `private static RectangleRoom[] rooms;`
 - (D) `private RectangleRoom[] rooms;`
 - (E) `private RectangleRoom[];`
30. Suppose now that you add a method called `getRooms()` to the `MontrealApartment` class. Which of the following snippets of code will let you call the method `getRooms` from another class without any compiler or runtime errors?
- I. `MontrealApartment ma; ma.getRooms();`
 - II. `MontrealApartment ma = new MontrealApartment(); ma.getRooms();`
 - III. `MontrealApartment.getRooms();`
- (A) II
 - (B) I
 - (C) All will work.
 - (D) I, II
 - (E) III
31. Suppose that you wanted to add a method to the `MontrealApartment` class that would print a list of all of the rooms. The rooms are all stored in an array called `rooms`. Which of the following statements could be used to access the name of the first element of the `rooms` array?
- (A) `rooms.name`
 - (B) `rooms[0].name`
 - (C) `rooms.name[0]`
 - (D) `RectangleRoom.name`
 - (E) `rooms[name][0]`
32. Suppose that the `MontrealApartment` class has a private property `price` representing the monthly rent price of the apartment. We want to write a method to let other classes access this private property. What would this method look like (inside the `MontrealApartment` class)?
- (A) `public static double getPrice() { return price; }`
 - (B) `private static double getPrice() { return price; }`
 - (C) `public double getPrice() { return price; }`
 - (D) `public void getPrice() { return price; }`
 - (E) `private double getPrice() { return price; }`
33. Suppose we want to write a method inside the `MontrealApartment` class that will calculate the total square footage of a `MontrealApartment`.
What should the header of this method be?

- (A) `private String getSquareFootage()`
(B) `private static int getSquareFootage()`
(C) `public int getSquareFootage()`
(D) `public static void getSquareFootage()`
(E) None of the above will compile because a get method can only return a property of the class.
34. Suppose we want to write a method inside the `MontrealApartment` class that will calculate the total square footage of a `MontrealApartment`.
How can we get the square footage of the first `RectangleRoom`?
- I. `rooms[0].getSize()`
 - II. `rooms[0].width * rooms[0].height * rooms[0].length`
 - III. `RectangleRoom.getSize()`
- (A) II
(B) I, II, and III
(C) II and III
(D) I
(E) I and II
35. Suppose we want to define a class to represent a `Balcony`. A `Balcony` does not have a height because it is outside and so the height is always the same (infinite). A `Balcony` has a width and a length. What is the minimum number of variables/properties you could define in order to represent the dimensions of the `Balcony` in memory.
- (A) 1
(B) It's impossible!
(C) 4
(D) 3
(E) 2

Bubble Sort

This section of the final is on bubble sort algorithm that you can find an implementation for in the appendix.

36. How many **swaps** will you have to do in order to sort this following array using bubble sort. Let n represent the number of elements in the array (9 in this case).
- ```
int[] a={ 1,2,4,3,5,6,7,8,9 };
```
- (A)  $n^2$   
(B)  $n-1$   
(C) 0  
(D) 1  
(E) 2
37. How many lines of code would you need to change in order to make *bubble sort* sort the entry values in decreasing order (rather than version presented which is increasing order)?

- (A) 2
  - (B) all of them
  - (C) impossible
  - (D) 0
  - (E) 1
38. What is the condition that is used to know if the array is sorted? Please pay attention to the code in the appendix.
- (A) Impossible to know it runs to infinity
  - (B) One swapping was done in the previous iteration of the for loop
  - (C) No swapping was done in the previous iteration of the for loop
  - (D) Once  $n-1$  scans are done
  - (E) Once  $n^2$  operations are done
39. How many times do you have to **scan** the following array to know that it is sorted? Let  $n$  represent the number of elements in the array (9 in this case).
- ```
int [] a={ 1,2,4,3,5,6,7,8,9 };
```
- (A) 0
 - (B) 1
 - (C) n^2
 - (D) $n-1$
 - (E) 9
40. Select the array that would take the most **scans** of the array to sort. Note that a scan is defined as one pass through the array.
- (A) [6, 0, 1, 2, 3, 4, 5]
 - (B) [0, 2, 1, 4, 3, 6, 5]
 - (C) [1, 2, 3, 4, 5, 6, 0]
 - (D) [0, 1, 2, 3, 4, 5, 6]
 - (E) All of the same

Binary Search

This section of the final is on binary search algorithm that you can find an implementation for in the appendix.

41. How many comparisons do you need to make to determine if the number 5 is in the following array. We count a comparison as anything that compares the value of two variables or expressions. This includes comparisons in while loop and if statements. Please pay attention to the binary search implementation in the appendix.

```
int [] a={1,2,3,4,5,6,7,8,9}
```

- (A) 3
- (B) 0
- (C) 2
- (D) 1

(E) 4

42. How many comparisons do you need to make to determine if the number 1 is in the following array. We count a comparison as anything that compares the value of two variables or expressions. This includes comparisons in while loop and if statements. Please pay attention to the binary search implementation in the appendix.

```
int[] a={1,2,3,4,5,6,7,8,9}
```

(A) 2

(B) 5

(C) 3

(D) 8

(E) 4

43. How many comparisons do you need to make to determine if the number -1 is in the following array. We count a comparison as anything that compares the value of two variables or expressions. This includes comparisons in while loop and if statements. Please pay attention to the binary search implementation in the appendix.

```
int[] a={1,2,3,4,5,6,7,8,9}
```

(A) 1

(B) 9

(C) 10

(D) Impossible

(E) 8

44. Since we used the decreasing bubble sort algorithm of question 37 on our array `a`. We would like to use our implementation of binary search to find index of different values. How many lines of code would you have to change to the binary search code in the appendix so it can search decreasing sorted array?

```
int[] a={9,8,7,6,5,4,3,2,1}
```

(A) All of it

(B) 2

(C) 1

(D) 10

(E) Impossible

Coding Section

In this section you are to write “methods” that use multiple questions, one line per question. The combination of questions can then form the whole method. **Make sure you explore all of the questions that define one method before deciding of your answer, as the obvious first choice might not be the right one.**

Also, do not forget the documentation in the appendix at the back as it may help for some of these questions.

Coding Question - Backward string

You are to write a method that accepts a `String` as input and prints, on the **same line**, the `String` backwards. Note that your method should not use `Scanner`. For example if the input is “Baie-Comeau”, the output should be “uaemoC-eiaB” all typed on the same line. Which line of codes do you need to write in order to create a method that will be called from the **main method** of that same file to produce such output? Use the next 5 following questions to “write” your code.

45. Line 1

- (A) `public void printBackwards(String input) {`
- (B) `public static void printBackwards() {`
- (C) `public static void printBackwards(String input) {`
- (D) `public static boolean doSomething() {`
- (E) `public static void main(String[] args) {`

46. Line 2

- (A) `int lengthInput = input.Length();`
- (B) `int lengthInput = input.length() - 1;`
- (C) `int lengthInput = input.length;`
- (D) `int lengthInput = input.size();`
- (E) `int lengthInput = input.length();`

47. Line 3

- (A) `for (int i = lengthInput; i >= 0; i = i - 1) {`
- (B) `for(int i = lengthInput - 1; i >= 0; i = i - 1) {`
- (C) `for(int i = lengthInput-1; i < 0; i = i - 1) {`
- (D) `for(lengthInput = 0; i < lengthInput; i++) {`
- (E) `for(int i = 0; i < lengthInput; i++) {`

48. Line 4

- (A) `System.out.println(input); }`
- (B) `System.out.print(input.charAt(i)); }`
- (C) `System.out.println(input.charAt(lengthInput-i - 1)); }`
- (D) `System.out.println(input.charAt(i)); }`
- (E) `System.out.print(input); }`

49. Line 5

- (A) `int x = 0; }`
- (B) `input = "" }`
- (C) `System.out.println(input); }`
- (D) `}`
- (E) `System.out.println(input.charAt(0)); }`

Coding Question - Drawing a square

You are to write a method that takes as input the size of a square and then outputs a square of that size. For example, if the input value is 4, the output would be as below:

```
####
####
####
####
```

Which line of codes do you need to write in order to create a method that will be called from the **main method** of that same file to produce such output? Like in the previous question, you should NOT use Scanner in your method. Use the 6 following questions to “write” your code. Pay close attention to how your answers group together with each other.

50. Line 1

- (A) public static void outputSquare() {
- (B) public static void needASquare(int n) {
- (C) public static boolean doSomething() {
- (D) public static void squareMe(String input) {
- (E) public static void main(String[] args) {

51. Line 2

- (A) int n = s;
- (B) None of these statements are useful when paired with other questions, leave blank.
- (C) String s = “”;
- (D) String s = “####”;
- (E) int[][] = new int[n][n];

52. Line 3

- (A) for(int i = 0; i < n; i++) {
- (B) for(int i = 0; i < n*n; i++) {
- (C) do {
- (D) None of these statements are useful when paired with other questions, leave blank.
- (E) while(true) {

53. Line 4

- (A) if(i % n == 0 && i > 0) { s+= “\n” }
- (B) if(j == 0 || i == 0 || i == n || j == n) { System.out.print(“#”) }
- (C) System.out.print(“#”);
- (D) for(int j = 0; j < n; j++) {
- (E) System.out.print(“\n”);

54. Line 5

- (A) System.out.print(“\n”);
- (B) s += “#”; }
- (C) System.out.print(“#”);

- (D) `System.out.println("");`
- (E) `}`

55. Line 6

- (A) `s += "#"; }`
- (B) `System.out.println(s); }`
- (C) `}`
- (D) `}} }`
- (E) None of these statements are useful when paired with other questions, leave blank.

Coding Question - Average calculation

You are to write a method that takes as input an array of integers and returns the average of that array. We are going to assume that we always receive a non-empty array, *i.e.* `array.length > 0`. The average of a set of numbers, a , is defined as follow,

$$avg = \frac{1}{n} \sum_{i=0}^n a_i = \frac{1}{n} (a_1 + a_2 + \cdots + a_n) \quad (1)$$

(The formula is a fancy way of specifying that one should add all the numbers together and divide the result by the total number of elements in the array.)

Which line of codes do you need to write in order to create a method that will be called from the **main method** of that same file to produce such output? Use the 6 following questions to “write” your code.

56. Line 1

- (A) `public static void main(String[] args) {`
- (B) `public void calculateAvg(int n) {`
- (C) `public static void avg(String input) {`
- (D) `public static boolean doSomething() {`
- (E) `public static double avg(int[] a) {`

57. Line 2

- (A) `for(int i = 0; i < a.length; i++) {`
- (B) `for(int i = 0; i <= a.length; i++) {`
- (C) `int sum = 0;`
- (D) `int sum = 0.0;`
- (E) None of these statements are useful when paired with other questions, leave blank.

58. Line 3

- (A) `for(int i = a.length; i > 0; i++) {`
- (B) `for(int i = 0; i < a.length; i++) {`
- (C) `for(int i = a.length-1; i > 0; i++) {`
- (D) `for(int i = 0; i <= a.length; i++) {`
- (E) `while(true) {`

59. Line 4

- (A) `sum += i; }`
- (B) `sum += a[i-1]; }`
- (C) `return a/a.length; }`
- (D) `sum += i + a[i]; }`
- (E) `sum += a[i]; }`

60. Line 5

- (A) `return (double) (sum / a.length;) }`
- (B) `return i; }`
- (C) `return ((double) sum)/ a.length; }`
- (D) `return a/a.length; }`
- (E) `return sum/a.length; }`

True false questions (1 point each)

In the following questions answer true or false

61. What will the following print?

```
int[] x = {1,2,3};
int[] y = new int[3];
y[0] = 1;
y[1] = 2;
y[2] = 3;
System.out.println(x == y);
```

- (A) false
- (B) true

62. An `ArrayList` is built into Java, which means you do not have to use an `import` statement.

- (A) TRUE
- (B) FALSE

63. In Java, if you have 5 public classes in a program you must create at least 5 files. (Note: For the purpose of this question we will not consider nested classes as they were not discussed in class.)

- (A) TRUE
- (B) FALSE

64. In Java, an `import` statement can be used to specify that you plan to use a certain type in your program.

- (A) FALSE
- (B) TRUE

65. In Java, if you have more than one class, it is required that only one of them have a main method.

- (A) FALSE
- (B) TRUE

66. When an `ArrayIndexOutOfBoundsException` occurs in a Java program, the program immediately terminates and no further statements can ever be executed.

- (A) TRUE
- (B) FALSE

67. The following code will crash when executed.

```
public static void main(String[] args) {  
    int[] x = {3};  
    try {  
        int y = x[10];  
    }  
    catch (ArrayIndexOutOfBoundsException e) {  
        int z = x[10];  
    }  
}
```

- (A) FALSE
- (B) TRUE

68. The primitive types such as `int` or `double` have no methods or properties defined on them in Java.

- (A) FALSE
- (B) TRUE

69. `String` is a reference type in Java.

- (A) FALSE
- (B) TRUE

70. The following code will first create a `String` `animal` with the value `goose` and then change it so that it has the value `moose`

```
String animal = "goose";  
animal.charAt(0) = 'm';
```

- (A) TRUE
- (B) FALSE

SUMMARY OF JAVA STANDARD LIBRARY METHODS FOR SELECTED CLASSES

• String (package java.lang) Methods:

- public boolean equals(Object anObject): Compares this String to anObject.
- public int length(): Calculates the length of this String.
- public char charAt(int i): Gets the char at position i of the String. Note that counting starts from 0 so that to get the first character of the String you should input i equals 0.
- public boolean equalsIgnoreCase(String anotherString): Compares, ignoring case considerations, this String to anotherString.
- public int compareTo(String anotherString): Compares this String to anotherString lexicographically; returns a negative value if this String occurs before anotherString, a positive value if this String occurs after anotherString, and 0 if both Strings are equal.
- public int compareToIgnoreCase(String anotherString): Compares, ignoring case considerations, this String to anotherString lexicographically; returns a negative value if this String occurs before anotherString, a positive value if this String occurs after anotherString, and 0 if both Strings are equal.
- public String substring(int start, int finish): Returns a new String composed of the this String starting from index start and up to, but not including index of finish
- public String replace(char c, char d): Returns a new String with all occurrences of the character c in the this String replaced by the character d.
- public char[] toCharArray(): Converts this String to a new character array.

• File (package java.io) Methods:

- public FileSString pathname(): Creates a new File instance that corresponds to the given pathname.

• Scanner (package java.util) Methods:

- public Scanner(InputStream source): Constructs a new Scanner that produces values scanned from the specified input stream.
- public Scanner(File f): Constructs a new Scanner that produces values scanned from the specified File
- public double nextDouble(): Scans the next token of the input as a double.
- public boolean nextBoolean(): Scans the next token of the input as a boolean.
- public int nextInt(): Scans the next token of the input as an int.
- public String nextLine(): Advances this Scanner past the current line and returns the input read.
- public boolean hasNextLine(): Checks whether there are further lines left to scan.

• PrintStream (package java.io) Methods:

- public void print(boolean b): Prints boolean value b.
- public void print(double d): Prints double value d.
- public void print(int i): Prints int value i.
- public void print(Object o): Prints Object o.
- public void print(String s): Prints Strings s.
- public void println(): Terminates the current line by writing the line separator string.
- public void println(boolean b): Prints boolean value b and then terminates the line.
- public void println(double d): Prints double value d and then terminates the line.
- public void println(int i): Prints int value i and then terminates the line.
- public void println(Object o): Prints Object o and then terminates the line.
- public void println(String s): Prints Strings s and then terminates the line.

• Math (package java.lang) Methods:

- public static double pow(double a, double b): Returns the value of a raised to the power of b.
- public static double sqrt(double a): Returns the correctly rounded positive square root of double value a.
- public static double random(): Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.
- public static double exp(double a): Returns Euler's number e raised to the power of double value a. (base e) of double value a. of double value a.

```
public void bubbleSort(int[] array) {
    int n = array.length;
    boolean madeSwap = true;
    while(madeSwap) {
        madeSwap = false;
        //Scanning
        for (int j = 0; j < n - 1; j++) {
            if (array[j] > array[j+1] ) {
                //Swapping
                int temp = array[j];
                array[j] = array[j+1];
                array[j+1] = temp;
                madeSwap = true;
            }
        }
    }
}

int binary_search(int[] array, int target, int min, int max)
{
    // continue searching while [min,max] is not empty
    while (max >= min)
    {
        // calculate the midpoint of min and max
        int mid = (max-min)/2;
        if(array[mid] == target)
        {
            // target found at index mid
            return mid;
        }
        // determine which subarray to search
        else if (array[mid] < target)
            // change min index to search upper subarray
            min = mid + 1;
        else
            // change max index to search lower subarray
            max = mid - 1;
    }
    // target not found. KEY_NOT_FOUND is an int constant defined.
    return KEY_NOT_FOUND;
}
```