

CISC 1115 Fall 2017 Sample Exam

Question 1.

Each of the following sections of code doesn't work properly. In each case, show how to fix the code so that it works properly. There is one error in each section of code. Put your answer in the box beneath the question.

a. The following section of code is supposed to calculate the sum of the numbers 1 to n. Assume that sum and n have been declared and n has been assigned a value.

```
for (int i = 1; i <= n; i++){
    sum = 0;
    sum = sum + i;
}
System.out.println("sum = " + sum);
```

b. The following section of code is supposed to print each of the numbers from 1 to n, each on a new line. (Assume that n has been assigned a value.)

```
for (int i = 1; i <= n; i++)
    System.out.println(n);
```

c. The following section of code is supposed to call the method doIt repeatedly as long as the user wants to continue.

```
char answer;
Scanner kybd = new Scanner(System.in);
do {
    doIt();
    System.out.print("Do you want to continue? ");
    answer = kybd.next().charAt(0);
} while (answer == 'y' && answer == 'Y');
```

d. The following method is supposed to check whether any of the values in the array are negative.

```
public static boolean hasNegValue(int[] a, int n)
{
    for (int i = 0; i < n; i++)
        if (a[i] < 0)
            return true;
        else
            return false;
}
```

Question 2.

Part A.

Given the following method:

```
int methodA(int x, int y) {  
    if (x < y)  
        return -1;  
    else if (x == y)  
        return 0;  
    else  
        return 1;  
}
```

What would be returned for the following calls:

- i. `methodA(3, 3)` _____
- ii. `methodA(5, 4)` _____
- iii. `methodA(20, 30)` _____
- iv. In general, what does the method return?

- v. Are there any other cases to take into consideration that are not covered by the above three calls? Answer either "No", or provide such a case.

- vi. Rewrite the method eliminating the if statements; instead use conditional operators only

Part B.

Given the following method:

```
String restOf(String s) {  
    return s.substring(1);  
}
```

What is returned by each of the following expressions:

- i. `"J" + restOf("Hello")` _____

- ii. `restOf(restOf(restOf("cowboy")))` _____

Question 3.

Part A:

A String variable sentence consists of words separated by single spaces.

Write the Java code needed to print the number of words that start with an upper case letter. So for "The deed is done", your code would print 1 but for "The name is Bond, JAMES Bond", your code would print 4.

Part B:

An int variable n contains a positive integer and a String variable s contains a String. Write the Java code needed to print "YES" if the first n characters of s appear more than once in s, and print "NO" otherwise.

Part C:

Show how to invoke the method grunf so that it will return the String HOPE.

```
public static String grunf(String s1, String s2, String s3) {
    String s = s1.substring(1,2);
    if (s1.length()>s2.length() && s2.compareTo(s3)<0
        && s3.indexOf(s)>0)
        return s1.substring(0,1) + s2.substring(1);
    else
        return s1.substring(0,1) + s2.substring(0,1) +
            s1.substring(0,1);
}
```

Question 4.

a. Suppose an exam grade called **grade** (integer) is considered **invalid** if it is either negative or above 100. Write code in Java that will set the variable **invalid** to true if the value of **grade** is invalid and false otherwise.

b. Write a section of code in Java that will do the following: Generate 10 random numbers (integers) each between 1 and 10. Print the number of random numbers that are greater than 5.

c. Consider an array of 10 integers that contains the following values in increasing sequence:

3 15 25 43 50 76 88 102 110 120

Assume you wish to find the location of the element with value 43.

i. Show the sequence of numbers that you would encounter in this search if you used a **sequential search**.

ii. Show the sequence of numbers that you would encounter in this search if you used a **binary search**.

d. Suppose you wish to compute the value of $a^2 + b^3$ for some integers a and b . Write a statement in Java that would place the value of this expression in an integer variable x .

Question 6.

Perform the following conversions:

a. 111001 (base 2) to base 10 Answer: _____

b. C4 (base 16) to base 10 Answer: _____

c. 1101001 (base 2) to base 16 Answer: _____

d. 34(base 10) to base 2 Answer: _____

e. E8 (base 16) to base 2 Answer: _____

f. Perform the following addition of two binary (base 2) numbers. Your answer should be a number in binary.

101 + 111 = ? Answer: _____

Question7.

For this question, assume all input comes from the keyboard, and all output goes to the screen. Include method prototypes and comments. The array should have room for 100 integers.

Write a complete Java program, including at least one comment in the main program and one in each method, to do the following:

Write a main program which will call the methods described below.

(a) First the main program will read an integer (this integer is a parameter or header value) which it will call **k**. Then the main program will call the method **readdata()** (see part 1 below) to read a set of data containing **k** values into an integer array which the main program will call **x**.

(b) The main program will call the method **count5s()** (see part 2 below), sending the **x** array (and **k**).

(c) The main program will call the method **sortarray()** (see part 3 below) to sort the first **k** elements of the **x** array into descending order. The new values in the **x** array will be printed in the main program.

Details of the methods:

1. Write a method called **readdata()** which receives two parameters, an integer **n** and an integer array **arr**.

The method will be sent a value for **n**. The method will read a set of **n** data items into the **arr** array. The method will print the set of data values as they are being read in.

2. Write a method called **count5s()** which receives two parameters, an integer **n** and an integer array **v**. The method will count how many of the first **n** elements of the **v** array are greater than 5, how many are less than 5, and how many are equal to 5.

The method will print these values, with messages for each.

As an example: if the array holds the values 5 66 -4 5 1 -3, with **n** = 6: there is 1 value greater than 5, there are 3 values less than 5, and there are 2 values equal to 5.

3. Write a method called **sortarray()** which receives two parameters, an integer array **w** and an integer **n** giving the size of the array. The method will sort the first **n** elements of the **w** array into descending order.

As an example, assume that method starts with these values in the **w** array: 5 66 -4 5 1 -3, with **n** = 6. Inside the method, these values will be sorted into descending order; the array will now hold 66 5 5 1 -3 -4.

