

Information and Computer Science Department
Spring Semester 191
ICS 102 – Introduction to Computing I
Midterm Exam
Thursday, October 24, 2019
Duration: 130 minutes

Name:

ID#:

--	--	--	--	--	--	--	--	--

Section No.

Question #	Max Score	Score
1	27	
2	18	
3	30	
4	25	
Total	100	

Q01. a) Write the output of the following java expression. If there is runtime error or compiler error write down the respective error.

Java expression	Output
4 / 7 * (double)7/4	0.0
137 / 42 == 0 && 137 / 0 == 42	false
1 + 2 + "1 + 2" +1 + 2	31 + 212
(char) ('3' - '0' + 'A')	D
1/2*(15-7)	0
(boolean) (4 < 5)	true
(boolean) (4 < 5 < 6)	Compiler error

Q01. b) Write down the output (that is displayed on the screen after executing the code) for the following java code fragment (assume the required variables are declared correct)

Code fragment	Output
<pre>int a = 10, b = 20; if (a < b) { if (a > b) { System.out.println("HELLO"); } else { System.out.println("WELCOME"); } } }</pre>	WELCOME
<pre>String greet = "Hi"; String name = "Smedley"; String nickName = name.substring(0,4); if (nickName == name.substring(0,4)) System.out.println("has real nickname"); else if (greet + name == greet + nickName) System.out.println("no real nickname"); else System.out.println("hmmm...changed names?");</pre>	hmmm...changed names?
<pre>int sum = 0; int i = 0; while (i < 3) { sum = sum + i; i++; } System.out.print(i); System.out.print(" "); System.out.print(sum);</pre>	3 3
<pre>String S = ""; String T = ""; int i = 4; for (i = 1; i <= 3; i++); S = S + "!"; for (i = 1; i < 4; i++) T = T + "*"; System.out.print(S); System.out.println(T);</pre>	!***

<pre>String str1 = new String("KFUPM"); String str2 = new String("KFUPM"); if (str1==str2) System.out.println("Equal"); else System.out.println("Not Equal");</pre>	<p>Not Equal</p>
<pre>int i=0, j =0; if(i++==j++) System.out.println(i-- + j--); else System.out.println(++i + ++j);</pre>	<p>2</p>
<pre>int n,t,x=0,r; Scanner kb = new Scanner(System.in); try{ System.out.print("Enter an integer: "); n = kb.nextInt(); t = n; while(t!=0){ r=t%10; x+=r; t/=10; } System.out.println(x); } catch(InputMismatchException e){ System.out.println("Error!!"+e); }</pre>	<p>Input: 567</p> <p>Your answer:</p> <p>18</p>

Q02.

a) Convert the following java code fragment to nested for loops:

```
int num, run,i,sum=0, count=0;
Scanner kb = new Scanner(System.in);
run = 0;
while(run++<5){
    i=0;
    while (i<10){
        num = kb.nextInt();
        if(num%5==0){
            sum+= num;
            count++;
        }
        i++;
    }
}
```

```
int num, run,i,sum=0, count=0;
Scanner kb = new Scanner(System.in);
for(run=0; run++<5; ){
    for(i=0; i<10; i++){
        num = kb.nextInt();
        if(num%5==0){
            sum+ = num;
            count++;
        }
    }
}
```

b) Convert the following java code fragment to switch-case and your code should not have any loops.

```
int i,num;
Scanner kb = new Scanner(System.in);

num = kb.nextInt();
for(i=1;i<3;i++){
    if(i==num){
        System.out.println("You have pressed : "+i);
        break;
    }
}
```

```
int i,num;
Scanner kb = new Scanner(System.in);

num = kb.nextInt();
switch(num){
    case 1: System.out.println("You have pressed : "+1);
        break;
    case 2: System.out.println("You have pressed : "+2);
        break;
}
```

c) Convert the following java code fragment to if-else.

```
int i,num;
Scanner kb = new Scanner(System.in);

num = kb.nextInt();
switch(num) {
case 1: System.out.println("You have pressed "+1); break;
case 2: System.out.println("You have pressed "+2); break;
default: switch(num%3) {
        case 0: System.out.println("Default zero"); break;
        case 1: System.out.println("Default one"); break;
        }
}
```

```
int i,num;
Scanner kb = new Scanner(System.in);

num = kb.nextInt();
if(num==1)
    System.out.println("You have pressed "+1);
else if(num==2)
    System.out.println("You have pressed "+2);
else if(num%3==0)
    System.out.println("Default zero");
else if(num%3==1)
    System.out.println("Default one");
```

Q03. Write down a java program that prompts to the user the following 3 options:

- A. Compute and display the total surface area of a cone
- B. Compute and display the distance between two points
- C. Quit

If the user chooses 'A', your program will prompt for height (h) and radius (r). Then compute the total surface area as follows:

$$\text{Total surface area} = \pi rh + \pi r^2$$

If the user chooses 'B', your program will prompt for the values of co-ordinates (x_1, y_1, x_2, y_2) (NB. All the values are positive). Then, follow the equation to compute the distance between the two points:

$$\text{distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

If the user chooses 'C', you quit the program (i.e., return).

Your program should check if the inputs are valid (i.e., h,r, x1, x2, y1,y2 are positive values).

If user chooses, any other characters other than the above three characters, the menu will be shown again to provide a correct input. For the choices 'A' or 'B', once the computations are finished and the results are displayed, the menu should be shown again for user to choose the above options again.

Sample output:

```
A.      Compute and display the total surface area of a cone
B.      Compute and display the distance between two points
C.      Quit

Enter your choice: A
Enter height(h) : 1.5
Enter radius(r) : 1.2

The total surface area is : 10.178760197630929
A.      Compute and display the total surface area of a cone
B.      Compute and display the distance between two points
C.      Quit

Enter your choice: B
Enter height coordinates of point 1 : 1
2
Enter height coordinates of point 2 : 3
4

The distance between two points is : 2.8284271247461903
A.      Compute and display the total surface area of a cone
B.      Compute and display the distance between two points
C.      Quit

Enter your choice: C
```

```

import java.util.*;

public class showMenu {
    public static void main(String[] args){
        char ch = 0;
        Scanner kb = new Scanner(System.in);

        do{
            try{
                System.out.println("A.      Compute and display the total surface area of a cone");
                System.out.println("B.      Compute and display the distance between two points");
                System.out.println("C. Quit");
                System.out.print("\nEnter your choice: ");
                ch = kb.nextLine().charAt(0);
                switch(ch){
                    case 'A':
                        case 'a': double h,r,area;
                            do{
                                System.out.print("Enter height(h) : ");
                                h = kb.nextDouble();
                                if(h<0)
                                    System.out.println("Height(h) must be positive! Try again.");
                            }while(h<0);
                            do{
                                System.out.print("Enter radius(r) : ");
                                r = kb.nextDouble();
                                if(r<0)
                                    System.out.println("Radius(r) must be positive! Try again.");
                            }while(r<0);

                                area = Math.PI*r*h + Math.PI*Math.pow(r, 2);
                                System.out.println("\nThe total surface area is : "+area);
                                kb.nextLine();
                                break;
                    case 'B':
                        case 'b': double x1=0,x2=0,y1=0,y2=0, distance;
                            do{
                                System.out.print("Enter height coordinates of point 1 : ");
                                x1 = kb.nextDouble();
                                y1 = kb.nextDouble();
                                if(x1<0){
                                    System.out.println("x1 must be positive!!");
                                    continue;
                                }
                                if(y1<0){
                                    System.out.println("y1 must be positive!!");
                                    continue;
                                }
                                System.out.print("Enter height coordinates of point 2 : ");
                                x2 = kb.nextDouble();
                                y2 = kb.nextDouble();
                                if(x2<0){
                                    System.out.println("x2 must be positive!!");
                                    continue;
                                }
                                if(y2<0){
                                    System.out.println("y2 must be positive!!");
                                    continue;
                                }
                            }while(x1<0 || y1<0 || x2<0 || y2<0);
                                distance = Math.sqrt(Math.pow(x1-x2, 2)+Math.pow(y1-y2, 2));
                                System.out.println("\nThe distance between two points is : "+distance);
                                kb.nextLine();
                                break;
                }
            }
            catch(InputMismatchException e){
                System.out.println("Invalid input.. Try again");
            }
        }while(ch!='c' && ch!='C');
    }
}

```


Q04. As you know the primitive data type int can't hold values greater than 2,147,483,647. However, sometimes, we need to handle with larger values than 2,147,483,647, for example, the number of atoms in one gram of carbon, programmers may use String in place of int to handle such large values. For example:

```
String x = "537658123450";  
String y = "43562300450067538";
```

The mathematical operation (say adding two such large numbers) using these numbers can be done by using the knowledge of adding one digit at a time that you learned (probably) at grade school. Let us recall, how we used to add two numbers here. First, consider two numbers (e.g, 456 and 565): one on top of the other as shown in the figure. Second, add two digits of each number starting from right to left. When the digits sum up to a value greater than ten, you just consider the left most digit of the sum as result and the right most digit should be considered as carry digit that would be added to the next column. You continue this, up to there is no digit left to add. The following figure shows how does it work:

Carry →	1	1	1	
Number 1 →		4	5	6
Number 2 →		5	6	5
Result →	1	0	2	1

Write down a java program, that will follow the same technique described above, to add two large numbers. Note, you must store the number in String type variable. You can assume that the length of each number is same.

The sample input/output of your program should be as follows:

```
<terminated> addition [Java Appli  
Enter first number: 456  
  
Enter second number: 565  
The result is:1021
```

```

import java.util.Scanner;

public class addition {
    public static void main(String[] args)
    {

        String num1;
        String num2;

        String sum = "";
        int digitDifference;

        int digitSum, digitCarry=0;

        Scanner kb = new Scanner(System.in);

        System.out.print("Enter first number: ");
        num1 = kb.next();

        System.out.print("\nEnter second number: ");
        num2 = kb.next();

        // If the two numbers are not of same length. Ignore this for the respective question

        if(num1.length()>num2.length()){
            digitDifference = num1.length()-num2.length();
            for (int i=1;i<=digitDifference;i++)
                num2 = "0"+num2;
        }
        else if(num1.length()<num2.length()){
            digitDifference = num2.length()-num1.length();
            for (int i=1;i<=digitDifference;i++)
                num1 = "0"+num1;
        }

        for (int i = num1.length()-1;i>=0;i--){
            digitSum = digitCarry+Integer.parseInt(String.valueOf(num1.charAt(i)),10) +
                Integer.parseInt(String.valueOf(num2.charAt(i)));

            if((digitSum-9) > 0){
                sum = Integer.toString(digitSum).charAt(1)+sum;
                digitCarry =
                    Integer.parseInt(String.valueOf(Integer.toString(digitSum).charAt(0)));
            }
            else{
                sum = Integer.toString(digitSum).charAt(0)+sum;
                digitCarry = 0;
            }

        }
        if(digitCarry==0)
            System.out.println("The result is:" + sum);
        else
            System.out.println("The result is:" + Integer.toString(digitCarry)+sum);
    }
}

```

Appendix

`Scanner kb = new Scanner(System.in);`

`kb.nextInt():` reads integer

`kb.nextDouble():` reads double value

`kb.nextLine():` reads sentence

`kb.next():` reads word

`Integer.parseInt(String.valueOf(num1.charAt(i))):` Converts the character at ith location of String num1 to integer. Example: num1 = 123. The resultant value will be 2 as integer for i=1.

`Integer.parseInt(String.valueOf(Integer.toString(x).charAt(0))):` First converts the integer x to String and then gets the left most value of the String; then converts the left most value to integer. Example: x = 18. The resultant value will be 8 as integer.

`Integer.parseInt(str):` Converts the str to integer.

`String.charAt(index);`

`Integer.toString(value):` converts the integer type value to String type

`str1.equals(str2);`

`str1.equalsIgnoreCase(str2);`

`str1.compareTo(str2);`

`kb.nextLine.charAt(0);`

`(char) System.in.read();`

`String.indexOf(str);`