**ervice (String custID, String Sname, int quan):** A non-default constructor, to initialize all service ariables. Note that initially all services are not paid.

**isplayService()**: Displays the info of the service in one line in the following format:

*customerID*<in 10 column width, left justified>*name*<in 10 column width, left justified>**x***quantity* = price <with two decimal points>**SR**

Example:

*128562 Facialx 2 = 240.00 SR*

**. Create an application class Salon that contains the following:**

**a.** A method **printInvoice (String custID,Service[] S):void**

This method prints the invoice for a certain customer, having the received cusomer*ID* as follows:

1. Print the info of each service the customer requested.
2. Print the total price for all the customer's services.
3. If the customer is a VIP customer, a 15% discount should be made on the total amount, and the new total (after reduction) should be printed.
   Note:A VIP customer is the customer having an ID that starts with *'V'*.
4. Marks all services as paid for the customer.

**b.** A method **RemainingDepts( Service[] ser): Service[]**

This method receives an array of services and returns another array of services that contain only the non-paid services.

**c. In the main:**

1. Create an array of ***services*** that can hold up to 50 services.
2. Read the information of 2 services from the user and add them to the array.
   i. Read the name and quantity of the service from the user.
   ii. The customerID is a randomly generated number ranging from 0-10000
3. Add to the array 3 services having the following information (not read from the user):

| CustomerID | Service name | Quantity |
|---|---|---|
| 12644 | Manicure | 1 |
| V11355 | Massage | 2 |
| 12644 | Facial | 1 |

4. Read a customer ID from the user and print his invoice.
5. Find all the unpaid services and print them.
6. Print the number of services requested from the salon (without looping through the array)

**Notes:**

- Use the same variable names given in the UML and question description. You will lose marks if you code contains different names.
- Reuse code whenever possible.

**mple run:**

```
---jGRASP exec: java Salon

please enter the name and quantity of the service you want
Facial
2
please enter the name and quantity of the service you want
Manicure
1
Please enter the customerID you want the invoice for
12644
12644      Manicure  x1 = 80.00SR
12644      Facial    x1 = 120.00SR
total = 200.0
The unpaid services are:
38649      Facial    x2 = 240.00SR
7334       Manicure  x1 = 80.00SR
V11355     Massage   x2 = 500.00SR
The number of services requested from the salon is 5

----jGRASP: operation complete.
```

Lab#...5.2.3.....

INSTRUCTIONS:

1- Create a **folder** in the **desktop** with your full name **firstname_lastname**
2- The first three lines of your code should specify your **full name, your ID and your lab section or time** as comments.
3- **Use the same variable names in the UML and description**, and meaningful names for other variables.
4- Reuse code whenever possible.
5- When you finish ,**Compress** your folder then upload it to **LMS**.
6- You are not allowed to user any electronic device or any material belonging to the course.
7- The duration of the exam: **3 hours.**

Write a program that helps a salon's reception handle and serve the customers. Follow the instructions below to achieve the task.

1. **Create a class *Service* according to the following UML and description:**

| *Service* |
|---|
| -customerID: String<br>-name: String<br>-quantity: int<br>-price: double<br>-paid: boolean<br>**+numServices: int** |
| +Service (String custID, String Sname, int quan)<br>+displayService():void<br>+getters/setters as necessary |

- **customerID**: A String representing the ID of the customer.
- **name**: The name of the service.
- **quantity**: The number of times this service needs to be repeated. (Example: if the customer requested two massages the value of the attribute will be 2).
- **price**: The total price of the service. Which is equal to ( quantity × base price of the service). The base price of the services are as follows:

| Service | Base price |
|---|---|
| HairCut | 100 |
| Manicure | 80 |
| Facial | 120 |
| Massage | 250 |

- **paid**: a Boolean variable indicating if the service have been paid or not.
- **numServices**: a static variable, counting the number of services created.

**Q4: Write one or more Java statements:** [4 pts.] (1 + 3 )

1. Write one statement only to calculate z based on the following formula (x , y are integers )

$$z = \sqrt{x^2 + y^2}$$

Z = sqrt ( x * x + POW(y,2) )

0.5 0.5

2. Write Java statements to find and print the duplicate values of an array of integer values my_arr

*Assume that* ***my_arr*** *declared and initialized as below:*

```
int[] my_arr = {1, 2, 5, 5, 6, 6, 7, 2};
```

0.25 0.5

for (int i=0; i < my_arr.length-1 )

0.5 for(int j=i+1; j < my_arr.length ; j++)

0.25 if( my_arr[i] == my_arr[j] )

0.25 print( my_arr[i] );

**Q1: State if the following statements are True or False:** [3 pts. =0.75 each]

| Statement | True/False |
|---|---|
| We can specify array size by either declaring it as a constant or reading it from the user during program execution. | T ✓ |
| The continue statement skips the remaining statements inside the loop; and proceeds with the next iteration, if any. | T ✓ |
| In void method, we can use break statement to exit early. | F ✓ |
| If the number of required iterations is not known, and the loop should execute at least once, then we must use a ***while*** loop. | F ✓ |

do while

(1 found)

-0.75

**Q2: Find and correct errors in the following code (there are 4 errors):** [4 pts.] (0.5 find error, 0.5 correct the error)

**Note:** Do not remove any statement.

```
1. import static java.lang.Math.*;
2. import java.util.*;
3. public class errors {
4. static Scanner input = new Scanner (System.in);
5. public static void main(String[] args){
6.   String[] array = {"Ahmad","451","Ali","T34"};
7.   int array2[]= new int[5];
8.   boolean flag= false;
9.   int index =0, num, sum =0;
10.  int j = input.nextInt();
11.  int minimum = min(j,array[1]);
12.  System.out.println("Enter:");
13.  String search = input.next();
14.    while (!flag && index<=array.length){
15.      if (search.equals(array[index]))
16.          flag=true;
17.      index++;
18.    }
19.    while (num<5){
20.      array2[num]=input.nextInt();
21.      find_negative(array2[num]);
22.      num++;
23.    }
24.  }
25.  public static boolean find_negative(int x) {
26.    if (x<0)
27.        System.out.println("Error!, Negative Numbers");
28.    else
29.        System.out.println("Positive Numbers");
30.    }
31.  }
```

| Line No | Error | Correction |
|---|---|---|
| 11 | int minimum=min(j, array[1]) | min cannot take String<br>min(j, ParseInteger"451"=451 |
| 19 | (num<5) num | int num=0; (num<5) |
| 21 | wrong calling (array2[num]) | int x=array2[num] find_negative(x); |
| 25 | boolean → | void / there is no return |

## Q5: Write a complete program: [5 pts.]

Write a program that prompts the user to enter 5 positive integers, and then for every number entered determines if the number is prime or not. The program should work as follows:

- Read the numbers in the main method.
- Call a boolean user-defined method named *isPrime* that returns true if the number is prime, otherwise false.
  - A number is prime if it is divisible only by 1 and itself. (you should write the complete method in the program).
- Print in the main method the message "It is prime" or the message "Not prime".

Sample run:

```
Enter a positive integer:7
It is prime
Enter a positive integer:-5
It is prime
Enter a positive integer:8
not prime
Enter a positive integer:1
It is prime
Enter a positive integer:15
not prime
```

```
import java.util.*;
Public class Q5{
static Scaner Read=new Scanner(System.in);
Public static void main (String [] args ){

For(int i=0; i<5 ; i++ ){
```

**Q3: What is the output for the following code segments:** [4 pts.] (2+2)

*Notes:*

- ***Please use ~ to represent space.***
- ***Any extra output is -0.25***

**1.**

```
import static java.lang.Math.*;
public class Program1{
   public static void main (String args[]){
      double x = -3.5;
      int [] a = new int[10];
      int [] b = new int[10];
      b = a;
      b[5]= 6;
      a[5] = 10;
      System.out.println( b[5] + " " + a[5]);
      System.out.println(round(x));
      System.out.println(ceil(3.5));
   }
}
```

**Output**

```
10 10
-2.0
4.0
```

**2.**

```
public class Program2 {
   public static void dis() {
      System.out.println(" in method dis");
   }
   public static int encrypt(int w, char q){
      int m;
      if (q == 'r')
         m= w/7;
      else {
         m= w%9;
         dis();
      }
      return m;
   }

   public static int m(int i) {
      return i+1;
   }

   public static void main(String[] args) {
      int y =0;
      int j =7;
      char []list = {'t','r'};
      for (int i=0; i<list.length;i++) {
         y =encrypt(j,list[i]);
         System.out.println(y);
      }
      System.out.println(m(j));
   } //end main
} // end class
```

**Output**

```
in method dis
7
1
17
```

Student Id: 437201570 Section Number: 25168

**Question#1: For each statement there is a list of options. Choose the option (only one) that would be the valid one.**

**1. What is the output of the following code fragment?**

```
public class Test1 {
public static void main(String args[]) {
int arr[] = {10, 20, 30, 40, 50}, j=10;
for(inti=0; i<arr.length; i++)
System.out.print(" " + arr[j]);
  }
}
```

a) 10 20 30 40 50

b) Compiler Error

c) 10 20 30 40

d) Run Time Error.

**2. What is the output of the following code fragment?**

```
public class Test2 {
public static void main (String[] args)
{
int arr1[] = {1, 2, 3};
int arr2[] = {1, 2, 3};
if (arr1 == arr2)
System.out.println("Same");
else
System.out.println("Not same");
  }
}
```

a) Same.

b) Not Same.

c) Compilation Error.

d) Run Time Error.

**3. What is the output of the following code fragment?**

```
public class Test3 {
public static void main (String[] args)
{
int arr1[] = {1, 2, 3};
int arr2[] = {1, 2, 3};arr3;
if (arr1==arr3)
System.out.println("Same");
else
System.out.println("Not same");
  }
}
```

a) Same.

b) Not Same.

c) Compilation Error.

d) Run Time Error.

**4.What is the output of the following code fragment?**

```
public class Test4 {
public static void main (String[] args)
{
for (inti=0; i<2; i++)
  for (int j=i+1; j<i+2;j++)
    for (int k=0; k<j;k++)
System.out.print(i +" "+j + " "+(++k));
      }
      }
```

a) 0 1 01 2 01 2 1

b) 0 1 11 2 1

c) 0 1 0 1 2 0 1 2 1

d) 0 1 10 2 1 1 2 1 1 3 1 1 3 3

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
COMPUTER SCIENCE DEPARTMENT

| CSC 111: COMPUTER PROGRAMMING-1 | Quiz 2 | 1st Semester 1438-1439 |
|---|---|---|

**5.What is the output of the following code fragment?**

```
public class Test4 {
public static void main (String[] args)
{
int num = 2;
for(int index= 0; index<egArray.length;
index=index*2)
  System.out.print(index);
    } }
```

a) 2 4 6 8 10 13 5 7 9

b) 2 6 10 3 7

c) 2 2 2 2 2 2 2 2 2 2

d) Infinite loop.

**6. Fill in the blank in the following code fragment so that each element of the array is assigned twice the value of its index.**

```
int[] array = new int[10];
for ( int index=0; index <array.length;
index++ )
    {  _____  }
```

a) index = 2*index;

b) array[ 2*index ] = 2*index;

c) array[ index ] = 2*array[ index ];

d) array[ index ] = 2*index;

**Question#2: Fill in the blanks of the following code fragment so that the elements of the array are printed *in reverse order*, starting with the last element.**

```
int[] myArray = { 2, 4, 6, 8, 10, 1, 3, 5, 7, 9 };
for(int index= _____0_____ ; index<myArray.length; index++ )
System.out.print(myArray[index ] + "  " );
```

**Question #3: Write java statements (loop) to print the following pattern:**

1
2 3
4 5 6
7 8 9 10

```
for (int i = 1 ; i < 10 ; i++) {
for int (int j = 1 ; j < i ; j++)
System.out.println(" ");
System.out.print (i);
```

**...alse:** [3 pts. =0.75 each]

| Statement | True/False |
|---|---|
| 1- Logical operators take *numeric* and *boolean* values as operands. | T ✗ -0.75 |
| 2- The *JVM* translate the code into bytecode. | F ✓ |
| s.substring(3) will return a string from index 3 up till s.length() -1 | T ✓ |
| Program terminates with a run-time error message, if the user enters the input value ***5.8*** for the variable ***num*** in this statement ***num = console.nextInt();*** | T ✓ |

int num =

**Q2: Find and correct errors in the following code segments (1 error each): [4 pts.] (0.5 find e**
**0.5 correct the error)**

| Line | a) |
|---|---|
| 1 | `final int DAYS_OF_MONTH= 31;` |
| 2 | `Scanner input = new Scanner(System.in);` |
| 3 | `String nameOfMonth;` |
| 4 | `nameOfMonth=input.next();` |
| 5 | `if(nameOfMonth.equals("February"))` |
| 6 | `DAYS_OF_MONTH= 28;` |
| Error | line 5 |
| Correction or Reason | if (namOfmonth=="February") (==) → Right one |

| Line | b) |
|---|---|
| 1 | `String str="CSC111";` |
| 2 | `char c='S';` |
| 3 | `System.out.print(str.indexOf(c)` |
| 4 | `System.out.print(str.charAt(2.5)` |
| 5 | |
| Error | line 4 |
| Correction or Reason | charAt(int) charA(2) |

| Line | c) |
|---|---|
| 1 | `int num=520;` |
| 2 | `double x=92.67;` |
| 3 | `String str= "Computer Science";` |
| 4 | `System.out.printf("%d%.2f%s%n",num,str, x);` |
| Error | (line 4) .2f% not for string for table and Float |
| Correction or Reason | System.out.printf("%d.%05 .%2f%n", num, str, x); |

| Line | d) |
|---|---|
| 1 | `int x=5;` |
| 2 | `if(!x<2)` |
| 3 | `System.out.println(x);` |
| 4 | `else x++;` |
| Error | line 2 |
| Correction or Reason | if(!(x<2)) |

: What is the output for the following code segments: [4 pts.] (1+1.5+1.5)

*tes:*

- *Please use ~ to represent space.*
- *Any extra output is -0.25*

a-

```
boolean x=false;
int t=0, j=1, k=2;
x= ++t==j || k%2>0;
System.out.print(t+" "+x+" "+k);
```

**Output**

1~True~True

-0.25

b-

```
//Unicode 'A'=65, 'B'=66, 'C'=67
char inputChar = 'A';
switch (inputChar) {
  case 'A':
  case 'a':
   System.out.println(inputChar);
   inputChar=(inputChar=='A')?++inputChar:'Z';

   case 'Z':
   case 'z':
   System.out.println(inputChar);
       break;

   default:
   System.out.println(inputChar + " is not
found");
       break;
   }
```

**Output**

A

66

-1

c-

```
int x=1, y=3, z=3;
if ((--x == 1) && (y == 3)) {
   System.out.println("good " + z--);
}
else if ((x == 0) && (y == 3)){
   System.out.println("better" + --z);
}
else {
   System.out.println("excellent" + z++);

}
```

**Output**

excellent~3

-1.5

## Q4: Write one or more Java statements: [5 pts.] (1 + 2 + 2)

a- Write one statement that will multiply the value of double variable ***dnum*** by 6 then assign it to i variable ***inum.***

*Note: Assume that all variables are already declared.*

```
int inum=(int)(dnum)*6);
```

b- Write a ***switch*** to test the value of integer variable **day**, if it is 1, 2, 3, 4, or 5 print "Weekday" if it 7 print "Weekend", otherwise print "Invalid".

```
Switch(day){
case 1:
case 2:
case 3:
case 4:
case 5: System.out.println("Weekday");
Break;
case 6:
case 7: System.out.println("Weekend");
Break;
default:
System.out.println("Invalid");
Break;
}
```

c- Write an ***if/else*** statement that adds 1 to the variable ***minors*** if the variable ***age*** is less than 18, adds 1 to the variable ***adults*** if ***age*** is between 18 and 64, and adds 1 to the variable ***seniors*** and prints the char 'S' if ***age*** is 65 or older.

*Note: All variables are of type int, declared and initialized.*

```
System.out.println("Enter age");
age=rand.nextInt();
if(age<18){
minors=++minors;
}
else if(18<age<64)
{
adults=++adults;
}
Seniors=++Seniors;
}
else
{
System.out.println("'S'");
```

-0.25

-0.25

**Q5: Write a complete program: [4 pts.]**

Read three temperatures, if their sum is greater than 60 degrees, print the message: "*FIRE!! Temper reached 60 degrees*". If one of the entered temperature value is negative, print the message: "*Negative was entered*".

```
import java.util.*;

public class temperatures {

public static void main (String args[]) {
Scanner Read=new Scanner(System.in);
System.out.println("Enter 3 temperatures");

int tem=Read.nextInt();
int tem1=Read.nextInt();
int tem2=Read.nextInt();

int Sum=tem+tem1+tem2;

if (Sum>60) {

System.out.println("FIRE!! Temperature reached 60 degrees");
}
else if ((tem<0)||(tem1<0)||(tem2<0)) {

System.out.println("Negative value was entered");
}

}} // End main and class
```