OL Academy

ITCS107/114

Computer Programming II

Midterm Exam Revision

Exam 1

First Semester, 2024/2025

Question 1

Find the output of the following program

```
public class Student
{
   private String Name;
   private int id;
   private static int count=0;

   public Student( String N, int d)
   { Name=N; id=d; count++; }

   public void SetStudent(String N, int d)
   {     Name=N;  id=d;     }

   public void Update(String N)
   { Name=N; }

   public void Update (int d)
   {
      if (d>0)
         id=d;
   }

   public void Update ( )
   { count++; }

   public String getName ( )
   { return Name;   }

   public int getID( )
   { return id;  }

   public int getCount( )
   { return count; }
```

```
public class demo{
public static void main(String [] args)
{
Student A = new Student("stA",123);
Student [] list = new Student[3];

list[0] = new Student("stB", 101);
list[1] = new Student("stC", (list[0].getID()+1));

System.out.println(list[0].getCount());
System.out.println(list[1].getID());
System.out.println(A.getName());

list[1].Update(-10);
A.Update( list[0].getName() );

for(int i=0;i<2;i++)
  System.out.println(list[i].getID());

System.out.println(A.getName());
System.out.println(A.getCount());


}
}
```

Output:

Question 2 Consider the following class:

```
public class player {
private String name;    private int score;   private int age;

public player(String n, int s, int a ) {  name=n;   score=s; age = a;}
public void setPlayer( String n, int s) { name=n;   score=s ; }
public void setName(String n) {name =n;}
public void setScore(int s){ score=s;}
public void setAge(int a){age =a;}
public String getName ( ) { return name; }
public int getScore( ) { return score; }
public int getAge( ) { return age; }
```

Write a static method named **updatePlayer** that takes as a parameter an array of type player named **list**. The method should count and output the number of young players (age below 10) and increase their scores by adding 5 points.

Note: assume the array list is declared and populated with data in the main method.

Question 3

Write the definition of a class named **BookInventory** that includes the following private members:

- **capacity**: an integer representing the maximum number of books.
- **Titles**: an array of type String representing the titles of books.
- **noCopies**: an array of type integer to hold the number of copies for each book in the Titles array.
- **numItems**: an integer to represent the actual number of books saved in the Titles array.

Note: Titles and noCopies are parallel arrays, thus for a given book in Titles[i], the number of copies of this book is saved in noCopies[i].

Additionally, the class should include the following public methods:

1. A **constructor** with an integer parameter (maxNo). The constructor should initialize **numItems** to zero, set the value of **capacity** to **maxNo** and create **Titles** and **noCopies** as arrays of size **capacity.**

2 A method named **findBookIndex** that accepts string (T) as parameter. The method should return the index of the book of Title (T). If the book is not found, the method should return –1.

3 A method named **addItem** that accepts two parameters: T as String and C as integer. The method should add (T) and (C) to the end of the **Titles** and **noCopies** arrays, respectively, and increment numItems. The addition should be performed if the array is not full and if (T) is unique (i.e. T is not already found in the Titles array).

4 A method named **outOfStock** to print the titles of books with no copies (i.e. noCopies is zero).

Question 4

Write a Java program (One class with a main method) to do the following:
1- Ask the user to enter (n) that represents the length of the array. The length of the array must be an even number otherwise print an appropriate message.
2- Declare two arrays of size n of type integer and prompt the user to enter the values of the arrays.
3- Swap the second half of the first array with the first half of the second array.
4- Print the elements of both arrays after the swap.

Sample Input/output	
Enter n: 5 The number of elements must be even	Enter n: 4 Enter 4 integers for the first array: 1 2 3 4 Enter 4 integers for the second array: 5 6 7 8 Array1 Elements after swapping: 1 2 5 6 Array2 Elements after swapping: 3 4 7 8

1. Ask the user to enter (n) that represents the length of the array. The length of the array must be an even number otherwise print an appropriate message.

2 Declare two arrays of size n of type integer and prompt the user to enter the values of the arrays.

3 Swap the second half of the first array with the first half of the second array.

4 Print the elements of both arrays after the swap.

Exam 2

Second Semester, 2024/2025

Question 1

Find the output of the following program

```
public class Product {
private String name;
private double price;
private static int pC = 0;
private static double tP = 0;

public Product ()
{ name ="Gift"; price=50; pC++;}

public Product (String n, double p)
{ name = n; price = p; pC++;  tP+=p;}

public void applyDiscount(double p)  {
  if (p > 0 && p <= 100)
  {
     tP    = tP -  (price * (p / 100));
     price = price - (price * (p / 100));
   }
}

public void applyDiscount()
{ tP = tp - 10;  price = price - 10; }

public String getName( ) { return name; }

public double getPrice() { return price; }

public static int getPC() { return pC; }

public static double getTP() { return tP; }
}// end of class
```

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

Product [] items = new Product [3];

items[0] = new Product("Laptop", 600);

items[0].applyDiscount(Product.getPC()*50);

items[1] = new Product("Smartphone", 400);

items[1].applyDiscount();

items[2] = new Product();


for (int i=0; i< 3; i++)
{
System.out.println(items[i].getName() + ": " + items[i].getPrice());
}

System.out.println("Products: " + Product.getPC());

System.out.println("Prices: " + Product.getTP());

}

}
```

Output:

Question 2 Consider the following class:

```
public class Rectangle {
 private int length;     private int width;
 public Rectangle( )
   {   length = 0;         width = 0;     }
public Rectangle(int newLength, int newWidth)
  { length = newLength;   width = newWidth; }
public int getLength() {return length;}
public int getWidth() {return width;}
}
```

Write a static method named **checkRectangle** that takes as a parameter an array of type Rectangle named list. The method should check the array and output the count of square shapes and the count of rectangular shapes. A rectangle is defined as a square if its length and width are equal; otherwise, it is a regular rectangle.

Note: Assume the array list is declared and populated with data in the main method.

Question 3

Write the definition of a class named **stList** that includes the following private members:
ID: to represent an array of students' IDs in a class. Note that a student ID is 9-digit number, where the first four digits represent the entry year, for example: and ID 202412345 is for a 2024-student
numStudents: to represent the number of students registered in the class.

Additionally, the class should include the following public methods:

1. A default constructor to declare ID as an array of size 40 and initialize numStudents to zero.

2 **findStudent** method that accepts (newID) parameter. The method should search for newID in the array and return its index. If the newID is not found, the method should return –1.

3 **addStudent** method that accepts (newID) as parameter. The method should add newID to the end
of the array ID, if:
1. ID array is not full
2. newID is a 2024 student
3. newID is not already added into the array

4 **print** method to print the numStudents and the ID array.

Question 4

Write a Java program (One class with a main method) to do the following:
1- Ask the user to enter two positive integers (m and n) to represent the lengths of two arrays.
2- Declare two arrays Ary1 and Ary2 (of size m and n respectively)
3- Prompt the user to enter the values of the two sorted arrays.
4- Merge the two sorted arrays Ary1 and Ary2 into a single sorted array Ary3, and print Ary3.

Sample Input/Output

```
Enter m: 4
Enter n: 5
Enter 4 integers for the first sorted array : 1 7 9 11
Enter 5 integers for the second sorted array: 2 4 6 8 20
Merged Sorted Array: 1 2 4 6 7 8 9 11 20
```

1 Ask the user to enter two positive integers (m and n) to represent the lengths of two arrays.

2 Declare two arrays Ary1 and Ary2 (of size m and n respectively)

3 Prompt the user to enter the values of the two sorted arrays.

4

Merge the two sorted arrays Ary1 and Ary2 into a single sorted array Ary3, and print Ary3.