وقت المراجعة

حضوريًا عن بعد

اليوم الثلاثاء

23/12/2025

07:00 PM

09:00

OL Academy

مدرس المقرر

أحمد كريم

ITCS106/113

Computer Programming I

Final Exam Revision

سجّل الآن
عبر موقعنا

Past Final Exam

Second Semester, 2024/2025

QUESTION #1:

Please Choose only one answer:

1 **What is the output of the following java code?**

```
int i=10;
do {
      if (i%2==1)
          continue;
      System.out.print(i);
}
while (i-->0);
```

a. 1086420

b. 108642

c. 97531

d. 975310

QUESTION #1:

Please Choose only one answer:

2 **What is the output of the following java code?**

```
public static int method1(int x){
    return x*3;
}

public static double method2(double x){
    return x*3+5;
}

public static void main(String[] args)
{

    double n = method2(3)-method1(2);
    System.out.println(n);
}
```

a. 1.0

b. 8.0

c. -2.0

d. 9.0

QUESTION #1:

Please Choose only one answer:

3 **What is the output of the following java code?**

```
String URL = "http://www.uob.edu.bh";
System.out.print(URL.substring(6,12));
System.out.print(URL.indexOf("www"));
```

a. //www.u8

b. //www. 8

c. /www. u7

d. /www.uo7

QUESTIN #1:

Please Choose only one answer:

4 **What is the output of the following java code?**

```
if ("Exam".compareTo("Test")>0)
    System.out.print("A");
if ( 'D' < 'X' )
System.out.print("B");

if ("25".compareTo("3")<0)
    System.out.println("C");
else
    System.out.print("D");
```

a. BC

b. ABC

c. B

d. A

QUESTIN #1:

Please Choose only one answer:

5 **What is the output of the following java code?**

```
public static int editMatrix(int[][] m){
    for(int i=0;i<=1;++i)
    m[i][i] *=2;
    return m[1][0]*3;
}
public static void main(String[] args) {
    int[][] matrix = {{2,1},{2,1}};
    int k = editMatrix(matrix);
    for (int x=0;x<=1;++x)
       for (int y=0;y<=1;++y)
           System.out.print(matrix[y][x]);
    System.out.print(k);
}
```

a. 41226

b. 21216

c. 212112

d. 42126

QUESTIN #1:

Please Choose only one answer:

6 **What is the output of the following java code?**

```
int[] x = new int[6];
x[0] = 3;
x[1] = x[0]*2;
for(int i=2;i<x.length;++i)
    x[i] = x[i-1]+x[i-2];
for(int i=4;i>=1;--i)
    System.out.print(x[i]);
```

a. 36915

b. 691524

c. 241596

d. 15963

QUESTION #2:

A car company exports new cars and wants to analyze the data on the cars they export to improve their operations. Write a Java program that prompts the user to input the number of cars exported and then asks for each car, the model name (String) and weight (double), and store them in two parallel arrays. Your program should do the following:

1. Find and display the heaviest car's weight and model name.
2. Find and display the lightest car's weight and model name.
3. Calculate and display the mean weight based on the following formula:

Mean weight = (heaviest car's weight} + lightest car's weight) / 2

4. Display the cars' model names that are heavier than the mean weight.

You program should follow the sample output below.

You program should follow the sample output below.

SAMPLE INPUT/OUTPUT

```
Enter the number of cars: 5
For Car 1: Enter Model's name and weight: ToyotaCamry      1400.5
For Car 2: Enter Model's name and weight: HondaAccord      1500.9
For Car 3: Enter Model's name and weight: NissanAltima     1463.7
For Car 4: Enter Model's name and weight: ChanganABC       1380.1
For Car 5: Enter Model's name and weight: Mazda6          1415.9

Heaviest car: HondaAccord, Weight: 1500.9 kg
Lightest car: ChanganABC, Weight: 1380.1 kg

Mean weight of the cars: 1440.5 kg

Cars' Model Names heavier than mean are:
 HondaAccord
NissanAltima
```

QUESTION #3:

Write a Java method named **airplaneCalc()**, that takes as input parameter a two-dimensional array named airport of size 4x5. By referring to the given Figure, the rows in the array airport represent the plane airline and the columns represent the days that the planes fly. The numbers in the cells of the array show the number of passengers that were on a given airline on a given day. The method will perform the following:

1. For each airline, calculate the total number of passengers and store them in the last column.
2. Display the total number of passengers for all airlines.
3. Find and display the lowest airline's total passengers with the airline number.

Airlines	Sun	Mon	Tue	Wed	
	0	1	2	3	Total
0	25	30	45	62	0
1	52	26	47	21	0
2	35	75	62	22	0
3	32	17	81	52	0

For example, if the main method for this program has the following array values as given below:

```
public static void main(String[]
args) { int[][] airport = {
{25, 30, 45, 62, 0},
{52, 26, 47, 21, 0},
{35, 75, 62, 22, 0},
{32, 17, 81, 52, 0}
};
airplaneCalc(airport);
}
```

Your **airplaneCalc()** method should display the following output:

```
Total passengers for airline 0 is 162
Total passengers for airline 1 is 146
Total passengers for airline 2 is 194
Total passengers for airline 3 is 182

Total passengers for all airlines is 684
Lowest total passengers is 146 for airline number 1
```

***You must use nested-loops to process the 2-dimensional array**

QUESTION #4:

Part 1: Define a class that represents a Robot in an industrial automation system with the following specifications:

a. The Class Name: Robot

b. Public Data Member:
   id (String): The unique ID of the robot.

c **Private Data Members**:

- **energyLevel** (int): The robot's remaining energy (default: 1000).
- **taskCapacity** (int): The number of tasks the robot can complete before requiring maintenance (default: 10).
- **tasksCompleted** (int): The number of tasks the robot has already completed (default: 0).

d **Constructor**: Accepts id, energyLevel, taskCapacity, and tasksCompleted as parameters.

- energyLevel must be between 500 and 2000.
- taskCapacity and taskCompleted must be positive.
- If any condition fails, set the data member to its default value.

e **Define**:

- An accessor (getter) method for **energyLevel**.
- A mutator (setter) method for **energyLevel**, ensuring it remains between 500 and 2000. Invalid value should not modify the attribute **energyLevel**.

f **Add a public method performTask(int energyRequired)**:

- Check if the robot has enough energy (**energyLevel >= energyRequired**). If not enough return false.
- Check if the **taskCompleted** hasn't exceeded its **taskCapacity**. If **taskCapacity** is exceeded, return false.
- If both conditions are met:
  - Reduce the **energyLevel** by **energyRequired**.
  - Increment **tasksCompleted** by 1.
  - Method should return true if that task is completed.

g **Add a public method recharge(int energy)**:

- Increase the robot's **energyLevel** by energy without exceeding 2000. If the **energyLevel** exceed 2000, set **energyLevel** to a value of 2000.

h **Add a public method printDetails()** to display the robot's ID, energy level, tasks completed, task capacity.

Part 2: Write a Java application

Write a Java application to do the following:

a Create a **Robot** object Robot1 with the following initial values:

ID	Energy Level	Task Capacity	Tasks Completed
R1001	500	10	3

b Make Robot1 perform a task requiring 300 energy.

c

Make Robot1 perform a task requiring 400 energy.
If task completed successfully, then Print: **"Task completed successfully by Robot [id]"**;
[id] is the robot id value.
If task is NOT completed, then recharge Robot1 with 200 energy.

d Print the details of Robot1 using the printDetails() method.

The Output of the Java Application for the above steps is shown below:

```
Robot ID: R1001
Energy Level: 400
Tasks Completed: 4
```