IQRA UNIVERSITY IU

North Campus

Subject	Object Oriented Programming (3+1)	**Program**	BS (CS)
Dated:	21_may_2023	**Deadline:**	
Student Name: **Student ID:**	Ahmer Aqeel 15960	**OBE:** CLO-2,3, PLO-2,3 and C2,3	
Assessment Activity	Assignment -2	**Max. Marks**	5
Semester	Spring 2023 (Semester- 3rd)	**Section Code**	

IMPORTANT INSTRUCTIONS:

Read the Instructions carefully.

1- Write your answers in a Word file and upload the file before the due date on Blackboard.

2- Write your name and registration ID on the first page of your Word file.

3- **Answer scripts can be uploaded on BB any time before its deadline. Therefore, do not wait for the last hour to avoid any unforeseen problems.**

4- **Submission of answer copy (ies) will be considered acceptable through BB only. Therefore, do not submit your document through email or any other medium.**

5- Use 12 pt. font size and Times New Roman font style along with 1-inch page margins.

6- Follow the requirements of the word limit and the marking criteria while writing your answers.

7- Provide relevant, original and conceptual answers, as this exam aims to test your ability to examine, explain, modify or develop concepts discussed in class.

8- Do not copy answers from the internet or other sources. The plagiarism of your answers ma be checked through Turnitin.

9- Recheck your answers before the submission on BB to correct any content or language relate errors.

10- Double check your word file before uploading it on BB to ensure that you have uploaded the correct file with your answers.

ASSIGNMENT TASKS:

1. **Apply** your **understanding** to write a JAVA program to print butterfly pattern as shown below

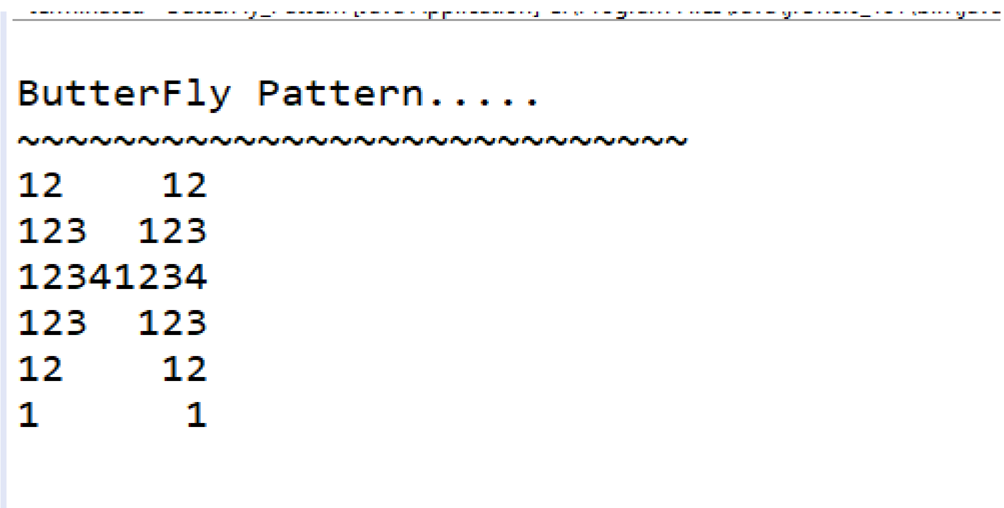
```
1 2        1 2
1 2 3  1 2 3
1 2 3 4 1 2 3
1 2 3  1 2 3
1 2        1 2
1            1
```

Source code:

```java
public class ButterFly_Pattern {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		System.out.println("\nButterFly Pattern.....");
		System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
		int n=4;

for(int i=2;i<=n;i++)
{
	for(int j=1;j<=i;j++)
	{

			System.out.print(j);
	}
	int gap=2*(n-i);
	for(int j=1;j<=gap;j++)
	{
		System.out.print(" ");
	}

	for(int j=1;j<=i;j++)
	{

			System.out.print(j);
	}
```

```
                        System.out.println();
}
for(int i=n-1;i>=1;i–)
{
      for(int j=1;j<=i;j++)
      {

                        System.out.print(j);
      }
      int gap=2*(n-i);
      for(int j=1;j<=gap;j++)
      {
            System.out.print(" ");
      }

      for(int j=1;j<=i;j++)
      {

                        System.out.print(j);
      }

                        System.out.println();
}
      }

}
```

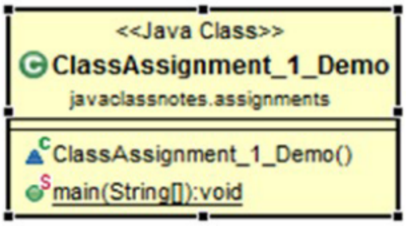
Output:

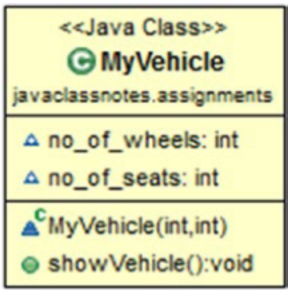


```
ButterFly Pattern.....
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
12    12
123  123
12341234
123  123
12    12
1      1
```

2. **Apply** your **understanding** to create a class Vehicle. The class should have two fields-no_of_seats and no_of_wheels. Create two objects-Motorcycle and Car for this class. Your output should show the descriptions for Car and Motorcycle.

   UML class diagram:



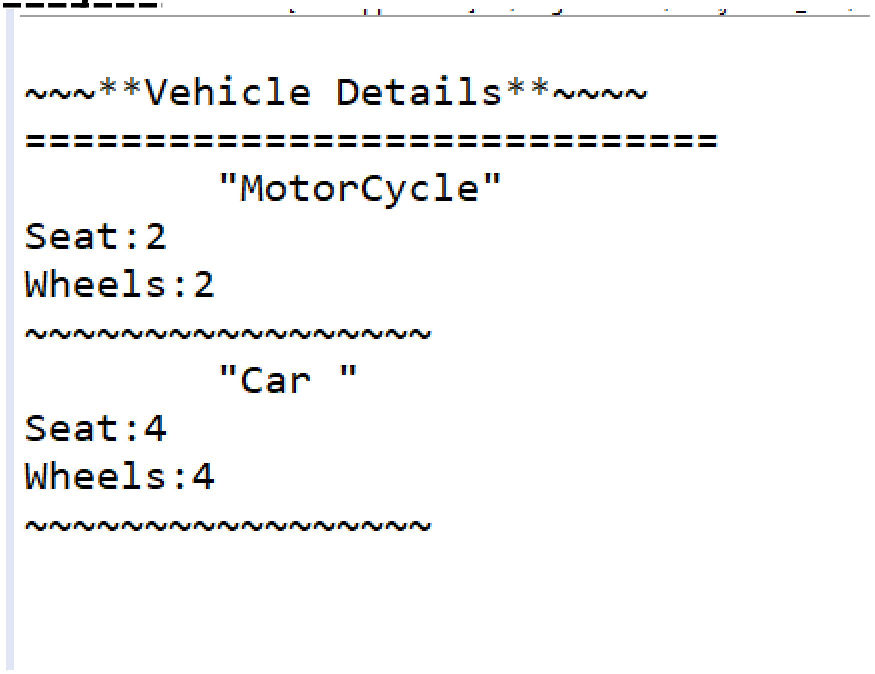


Source code:

```
public class My_Vehicles {
	int no_of_seat;
	int no_of_wheels;
	My_Vehicles(int no_of_seat,int no_of_wheels)
	{
		this.no_of_seat=no_of_seat;
		this.no_of_wheels=no_of_wheels;
	}
	public String toString()
	{
		return "\nSeat:"+no_of_seat+"\nWheels:"+no_of_wheels;
	}
	}
public class MainClass {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		My_Vehicles motorcycle=new My_Vehicles(2,2);
		System.out.println("\n~~~**Vehicle Details**~~~~");
		System.out.println("================================");
		My_Vehicles car=new My_Vehicles(4,4);
		System.out.println("\t\"MotorCycle\" "+motorcycle);
		System.out.println("~~~~~~~~~~~~~~~~~");
		System.out.println("\t\"Car \""+car);
		System.out.println("~~~~~~~~~~~~~~~~~");

	}
```

}

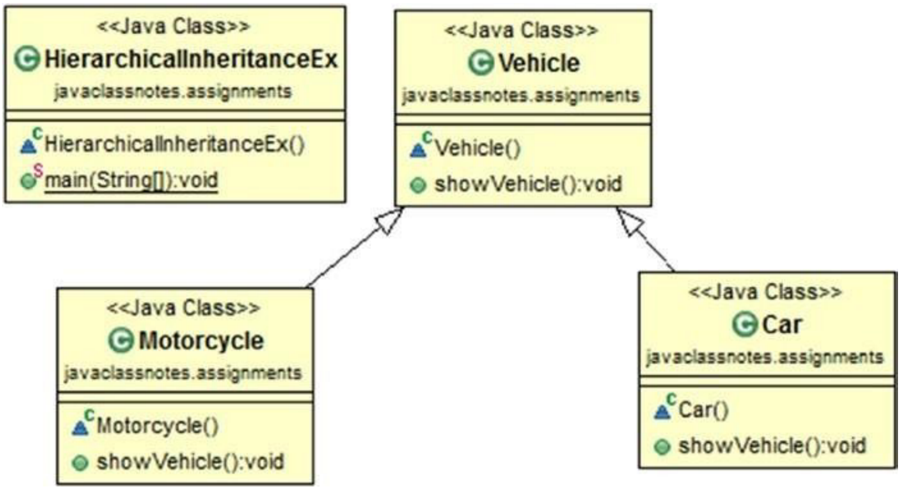
Output:



```
~~~**Vehicle Details**~~~~
=============================
        "MotorCycle"
Seat:2
Wheels:2
~~~~~~~~~~~~~~~~~
        "Car "
Seat:4
Wheels:4
~~~~~~~~~~~~~~~~~
```

3. **Apply** your **understanding** to write a simple program to implement hierarchical inheritance.

   UML class diagram:



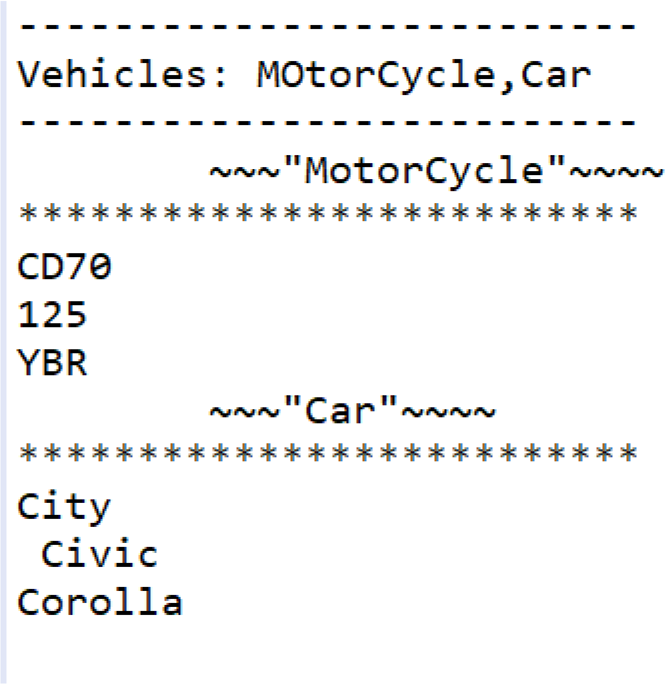
Source code:

```
public class Vehicle {
        Vehicle()
        {
```

```
	}
	void showVehicle()
	{
		System.out.println("-------------------------");
		System.out.println("Vehicles: MOtorCycle,Car");
		System.out.println("-------------------------");

	}

}


public class MotorCycle {
	MotorCycle()
	{
		System.out.println("\t~~~\"MotorCycle\"~~~~");
		System.out.println("**************************");
	}
	void showvehicle()
	{
		System.out.println("CD70\n125\nYBR");
	}

}
public class Car {
	Car()
	{
		System.out.println("\t~~~\"Car\"~~~~");
		System.out.println("**************************");
	}
	void showvehicle()
	{
		System.out.println("City\n Civic\nCorolla");
	}

}
public class Main_class {
	public static void main(String[] args) {
		// TODO Auto-generated method stub

	Vehicle v=new Vehicle();
	v.showVehicle();
```

```
        MotorCycle m=new MotorCycle();
        m.showvehicle();
        Car c=new Car();
        c.showvehicle();
        }
}
```

Output:



```
--------------------------
Vehicles: MOtorCycle,Car
--------------------------
        ~~~"MotorCycle"~~~~
**************************
CD70
125
YBR
        ~~~"Car"~~~~
**************************
City
 Civic
Corolla
```