Lab#02

Implementation of Objects and classes

Objective

- Classes
- Objects
- Transformation from Procedural to Object Oriented Programming
- Example Programs
- Exercise

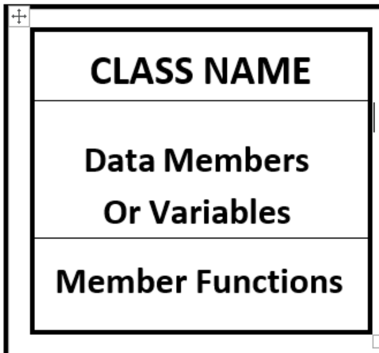
Theory

CLASS

A class is a programmer-defined data type that describes what an object of the class will look like when it is created. It consists of a set of variables and a set of functions.

Classes are created using the keyword class. A class declaration defines a new type that links code and data. This new type is then used to declare objects of that class.

In the UML, a class icon can be subdivided into compartments. The top compartment is for the name of the class, the second is for the variables of the class, and the third is for the methods of the class.



```
class class-name
{
access-specifier:
data

access-specifier:
functions

};
```

CLASS NAME

By convention, the name of a user-defined class begins with a capital letter and, for readability, each subsequent word in the class name begins with a capital letter.

DATA MEMBERS

Consider the attributes of some real-world objects:

RADIO – station setting, volume setting.

CAR – speedometer readings, amount of gas in its tank and what gear it is in.

These attributes form the data in our program. The values that these attributes take (the blue color of the petals, for example) form the state of the object.

MEMBER FUNCTIONS

Consider the operations of some real-world objects:

RADIO – setting its station and volume (invoked by the person adjusting the radio's controls)

CAR – accelerating (invoked by the driver), decelerating, turning and shifting gears.

These operations form the functions in program. Member functions define the class's behaviors.

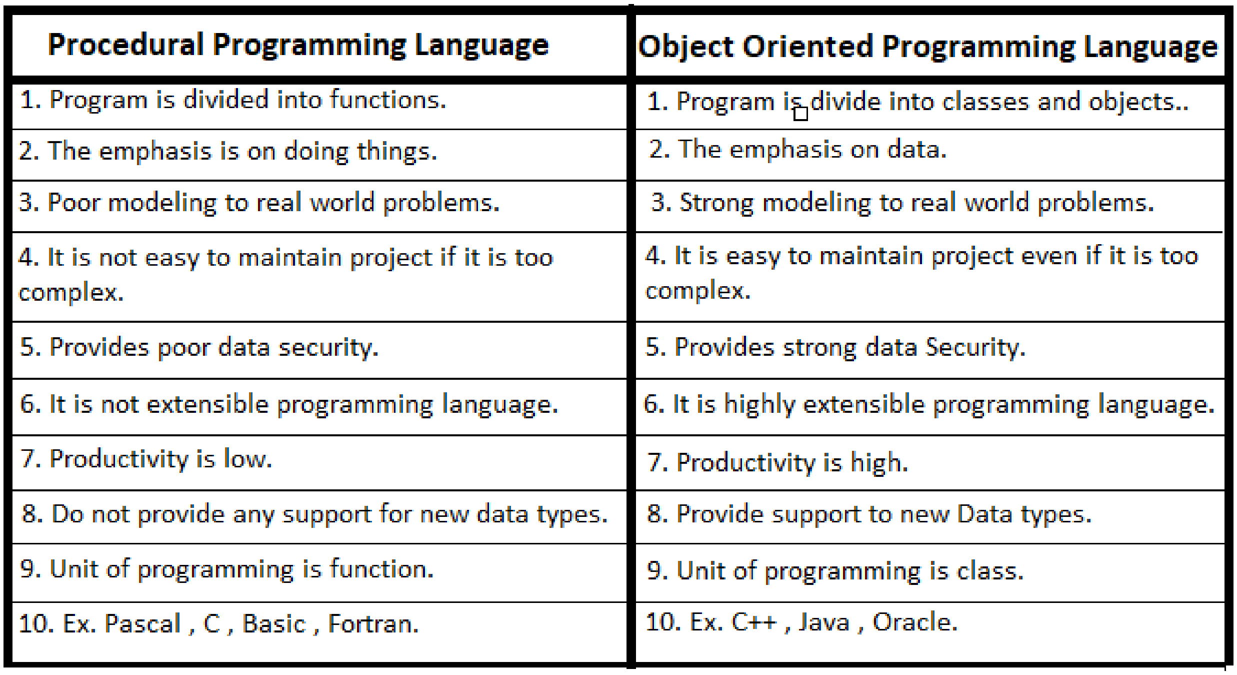
OBJECT

When we define a variable of a class, we call it instantiating the class. The variable itself is called an instance of the class. A variable of a class type is also called an object. Instantiating a variable allocates memory for the object.

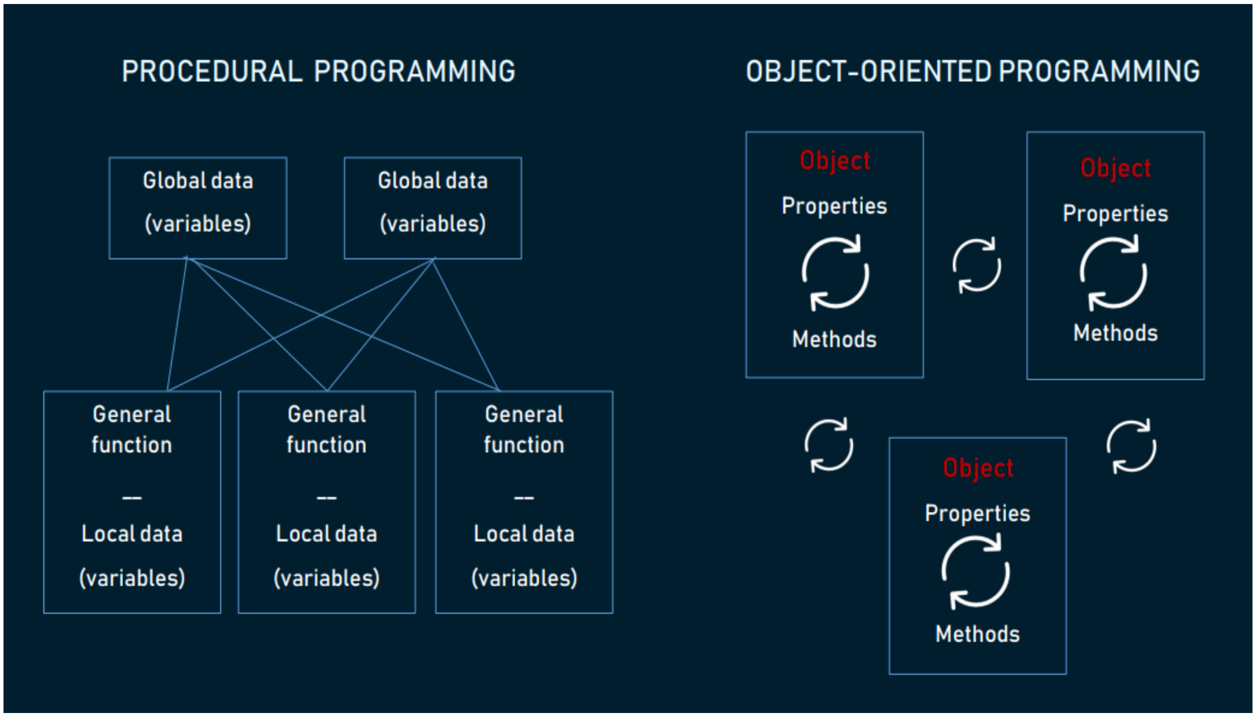
RADIO r;

CAR c

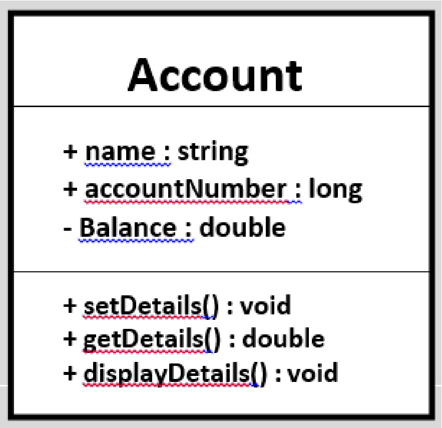
TRANSFORMATION FROM PROCEDURAL TO OBJECT ORIENTED PROGRAMMING



Procedural Programming Language	Object Oriented Programming Language
1. Program is divided into functions.	1. Program is divide into classes and objects..
2. The emphasis is on doing things.	2. The emphasis on data.
3. Poor modeling to real world problems.	3. Strong modeling to real world problems.
4. It is not easy to maintain project if it is too complex.	4. It is easy to maintain project even if it is too complex.
5. Provides poor data security.	5. Provides strong data Security.
6. It is not extensible programming language.	6. It is highly extensible programming language.
7. Productivity is low.	7. Productivity is high.
8. Do not provide any support for new data types.	8. Provide support to new Data types.
9. Unit of programming is function.	9. Unit of programming is class.
10. Ex. Pascal , C , Basic , Fortran.	10. Ex. C++ , Java , Oracle.



Example program:

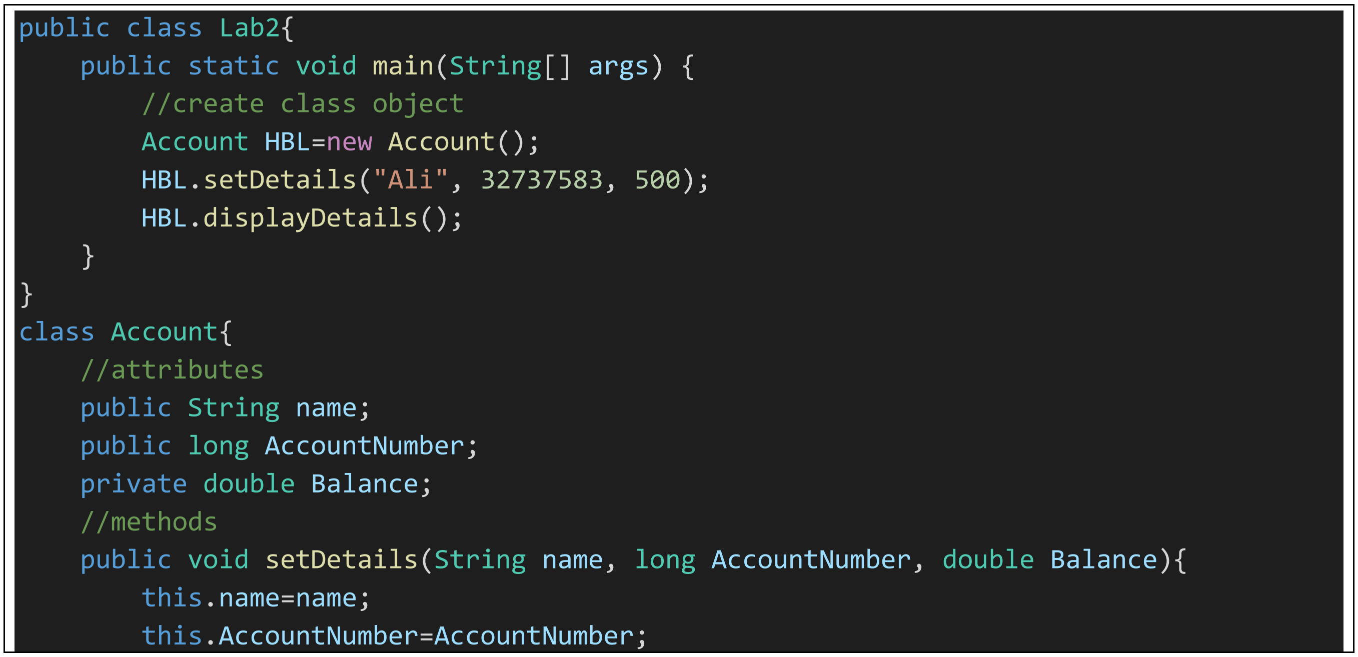


Account

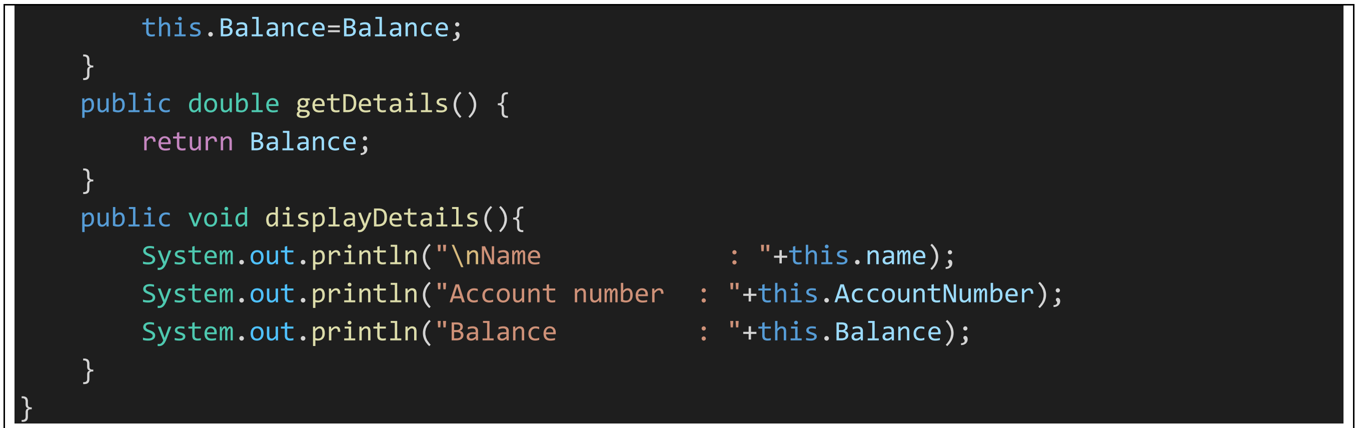
+ name : string
+ accountNumber : long
- Balance : double

+ setDetails() : void
+ getDetails() : double
+ displayDetails() : void

Code:

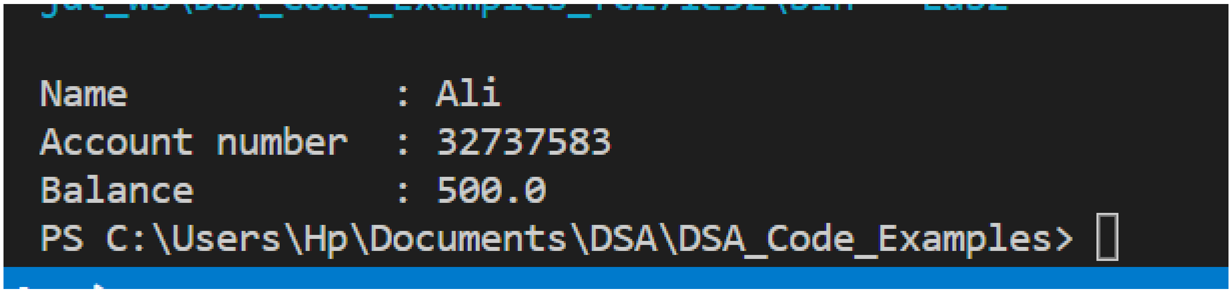


```
public class Lab2{
    public static void main(String[] args) {
        //create class object
        Account HBL=new Account();
        HBL.setDetails("Ali", 32737583, 500);
        HBL.displayDetails();
    }
}
class Account{
    //attributes
    public String name;
    public long AccountNumber;
    private double Balance;
    //methods
    public void setDetails(String name, long AccountNumber, double Balance){
        this.name=name;
        this.AccountNumber=AccountNumber;
```



```
        this.Balance=Balance;
    }
    public double getDetails() {
        return Balance;
    }
    public void displayDetails(){
        System.out.println("\nName             : "+this.name);
        System.out.println("Account number  : "+this.AccountNumber);
        System.out.println("Balance         : "+this.Balance);
    }
}
```

Output:



```

Name            : Ali
Account number  : 32737583
Balance         : 500.0
PS C:\Users\Hp\Documents\DSA\DSA_Code_Examples> 
```

Exercise

INSTRUCTIONS:

NOTE: Violation of any of the following instructions may lead to the cancellation of your submission.

1. Create a word file name it by your student rollNo(NOTE format is given on teams use that format)
2. Paste all the code along with the out output screenshot and for each question with the names such as Q1.java, Q2.java
3. Submit the word/pdf file on teams.

QUESTION#1

Create a class Rectangle with attributes length and width, each of which defaults to 1. Provide member functions that calculate the perimeter and the area of the rectangle. Also, provide set and get functions for the length and width attributes. The set functions should verify that length and width are each floating-point numbers larger than 0.0 and lessthan 20.0.

QUESTION#2

Create a class called Employee that includes three pieces of information as data members—a first name (type char*), a last name (type string) and a monthly salary (type int). Your class should have a setter function that initializes the three data members. Provide a getter function for each data member. If the monthly salary is not positive, set it to 0.Write a test program that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10 percent raise and display each Employee's yearly salary again. Identify and add any other related functions to achieve the said goal.

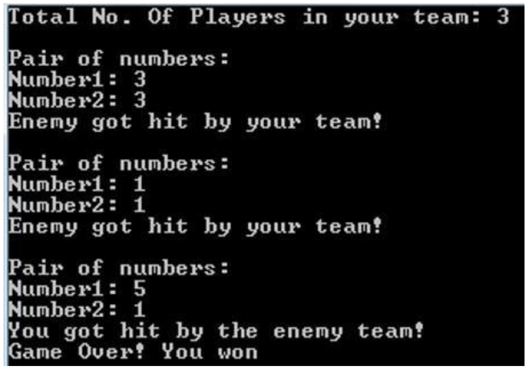
QUESTION#3

Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four data members—a part number (type string), a part description (type string), a quantity of theitem being purchased (type int) and a price per item (type float). Your class should have a functions that initializes the fourdata members. Provide a get function for each data member. In addition, provide a member function named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns theamount as a float value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0. Write a test program that demonstrates class Invoice's capabilities.

QUESTION#4

Write C++ code to represent a hitting game. The details are as follows:
This game is being played between two teams (i.e. your team and the enemy team).
The total number of players in your team is randomly generated and stored
accordingly.
The function generates a pair of numbers and matches each pair. If the numbers get matched, the following message isdisplayed:
"Enemy got hit by your
team!"Otherwise, the following message is displayed:
"You got hit by the enemy team!"
The number of hits should be equal to the number of players in your team.
The program should tell the final result of your team by counting the hits of both the
teams.

Consider the following sample output



```
Total No. Of Players in your team: 3

Pair of numbers:
Number1: 3
Number2: 3
Enemy got hit by your team!

Pair of numbers:
Number1: 1
Number2: 1
Enemy got hit by your team!

Pair of numbers:
Number1: 5
Number2: 1
You got hit by the enemy team!
Game Over! You won
```

QUESTION#5

MyJava Coffee Outlet runs a catalog business. It sells only one type of coffee beans. The company sells the coffee in 2-lb bags only and the price of a single 2-lb bag is $5.50 when a customer places an order, the company ships the order in boxes. The boxes come in 3 sizes with 3 different costs:

	Large Box	Medium Box	Small Box
Capacity	20 Bags	10 Bags	5 Bags
Cost	$1.80	$1.00	$0.60

The order is shipped using the least number of boxes. For example, the order of 52 bags will be shipped in 2 boxes: 2large boxes, 1medium and 1 small. Develop an application that computes the total cost of an order.

Number of Bags
Ordered: 52The Cost of
Order: $ 286.00 Boxes
Used:
2 Large - $3.60
1 Medium -
$1.001 Small -
$0.60
Your total cost is: $ 291.20

QUESTION#06

Write a class named Vehicle that can represent both the Rickshaw and Bike on the basis of number of wheels it has. Eachvehicle has the following details

- **year**. An int that holds the vehicle’s model year.
- **manufacturer**. A string that holds the manufacturer name of that vehicle.
- **speed**. An int that holds the vehicle’s current speed.

In addition, the class should have the following member functions.

- **accelerate**. The accelerate function should add 5 to the speed member variable each time itis called.
- **brake**. The brake function should subtract 5 from the speed member variable each time it iscalled.

Demonstrate the class in a program that creates a Vehicle object for a Rickshaw and for a Bike both, and then calls the accelerate function five times. After each call to the accelerate function, get the current speed of the car and display it. Then, call the brake function two times. After each call to the brake function, get the current speed of the car and displayit

Happy coding 😊