**CSE 215L: Programming language II Lab**
**Faculty: Dr. Mohammad Rashedur Rahman (RRn)**
**Lab - 06 [Introduction to OOP]**
**Fall-2022**
**Lab Instructor: Md. Mustafizur Rahman**

**Objective:**

- OOP in Java
- Class Definition in Java
- Creating Instances of a Class
- UML class and Instance Diagrams

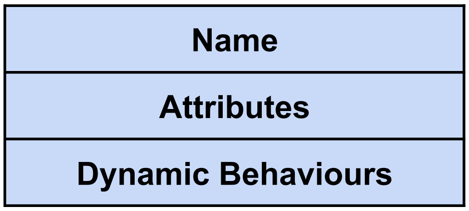
## Class & Instances

In Java, a class is a definition of objects of the same kind. In other words, a class is a blueprint, template, or prototype that defines and describes the static attributes and dynamic behaviors common to all objects of the same kind.
An instance is the realization of a particular item of a class. In other words, an instance is an instantiation of a class. All class instances have similar properties, as described in the class definition.

A class can be visualized as a three-compartment box, as illustrated:

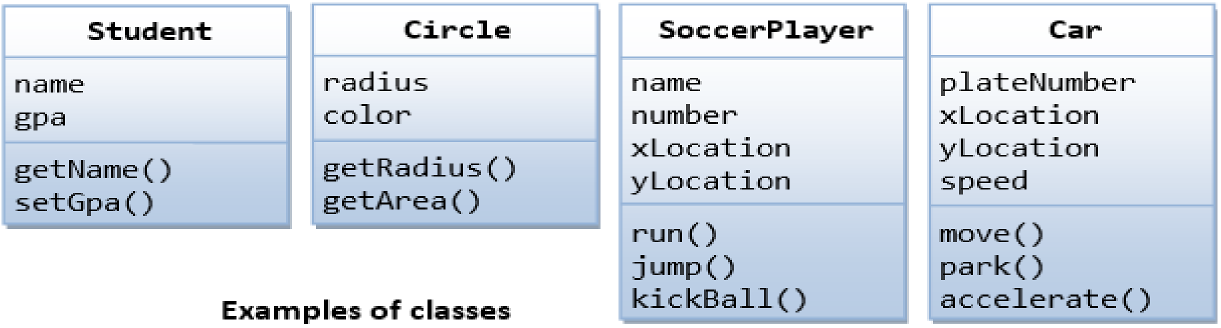
1. *Name* (or identity): identifies the class.
2. *Variables* (or attribute, state, field): contains the *attributes* of the class.
3. *Methods* (or behaviors, function, operation): contains the *dynamic behaviors* of the class.

The followings figure shows a few examples of classes:



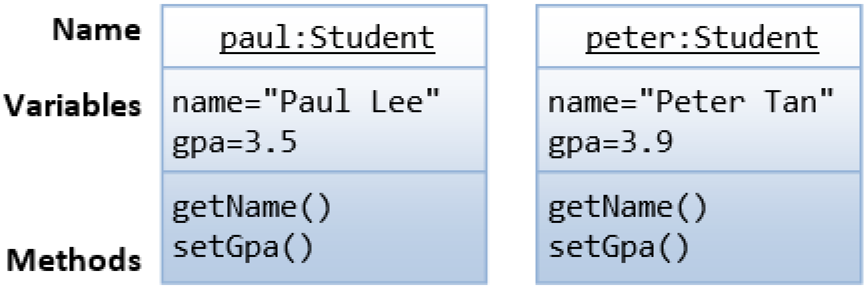
A class is a 3-compartment box

The followings figure shows a few examples of classes:



**Examples of classes**

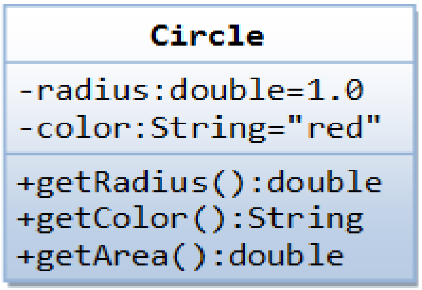
The following figure shows two instances of the class Student, identified as "paul" and "peter".



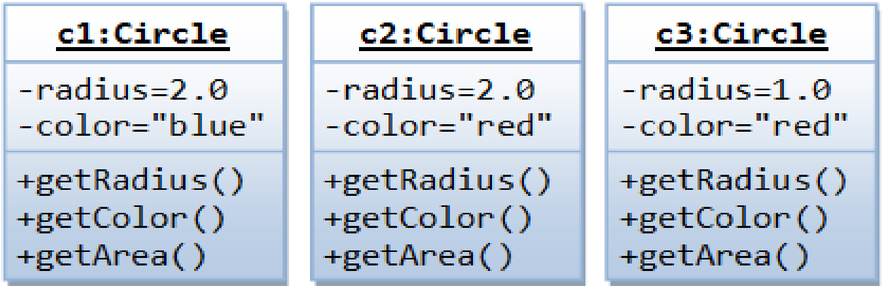
**Two instances - paul and peter - of the class Student**

## An OOP Example:

**Class Definition**



**Instances**



```
/*
 * The Circle class models a circle with a radius and color.
 */
public class Circle {    // Save as "Circle.java"
   // Private instance variables
   private double radius;
   private String color;

   // Constructors (overloaded)
   public Circle() {                   // 1st Constructor
      radius = 1.0;
      color = "red";
   }
   public Circle(double r) {           // 2nd Constructor
      radius = r;
      color = "red";
   }
   public Circle(double r, String c) { // 3rd Constructor
      radius = r;
      color = c;
   }

   // Public methods
   public double getRadius() {
      return radius;
   }
   public String getColor() {
      return color;
   }
   public double getArea() {
      return radius * radius * Math.PI;
   }
}
```

### TestCircle.java

We shall now write another class called TestCircle, which uses the Circle class. The TestCircle class has a main() method and can be executed.

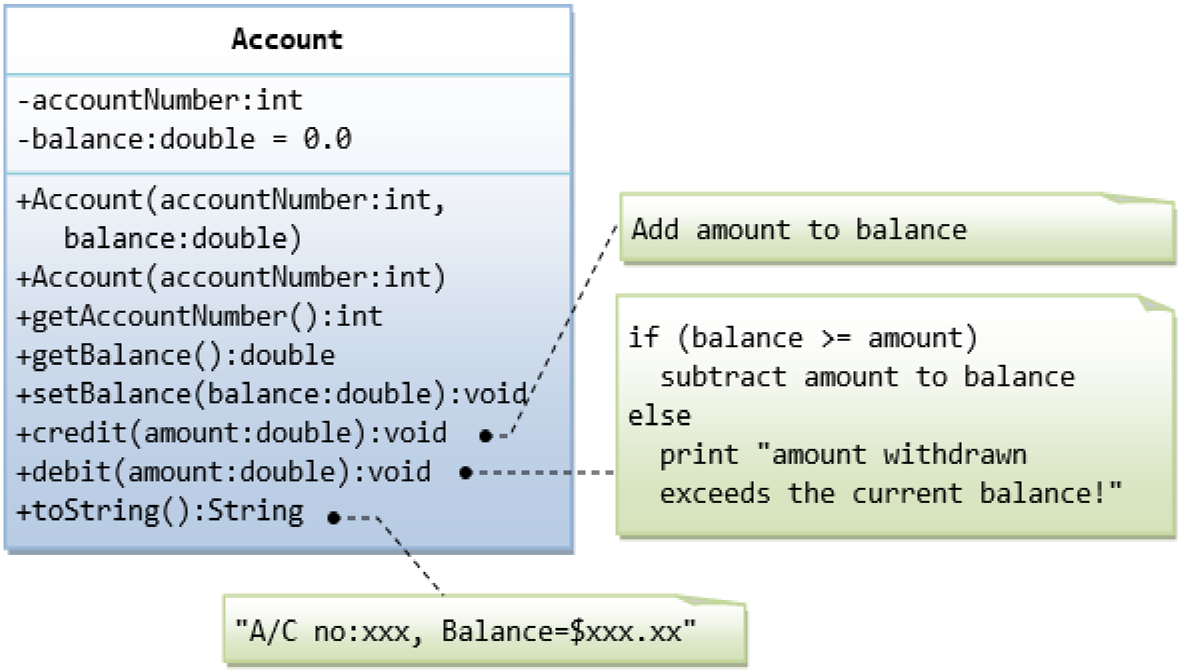
```
/*
 * A Test Driver for the "Circle" class
 */
public class TestCircle {    // Save as "TestCircle.java"
   public static void main(String[] args) {   // Program entry point
      // Declare and Construct an instance of the Circle class called c1
      Circle c1 = new Circle(2.0, "blue");  // Use 3rd constructor
      System.out.println("The radius is: " + c1.getRadius());  // use dot operator to invoke
member methods
      System.out.println("The color is: " + c1.getColor());
      System.out.printf("The area is: %.2f%n", c1.getArea());

      // Declare and Construct another instance of the Circle class called c2
      Circle c2 = new Circle(2.0);  // Use 2nd constructor
      System.out.println("The radius is: " + c2.getRadius());
      System.out.println("The color is: " + c2.getColor());
      System.out.printf("The area is: %.2f%n", c2.getArea());

      // Declare and Construct yet another instance of the Circle class called c3
      Circle c3 = new Circle();  // Use 1st constructor
      System.out.println("The radius is: " + c3.getRadius());
      System.out.println("The color is: " + c3.getColor());
      System.out.printf("The area is: %.2f%n", c3.getArea());
   }
}
```

Compile TestCircle.java into TestCircle.class.

**Task: Implement the following class and test its methods**


Account
-accountNumber:int
-balance:double = 0.0
+Account(accountNumber:int,
balance:double)
+Account(accountNumber:int)
+getAccountNumber():int
+getBalance():double
+setBalance(balance:double):void
+credit(amount:double):void
+debit(amount:double):void
+toString():String
Add amount to balance
if (balance >= amount)
subtract amount to balance
else
print "amount withdrawn
exceeds the current balance!"
"A/C no:xxx, Balance=$xxx.xx"