CSE 215L: Programming Language II Lab
Section: 7
Fall 2020

Today's topics:

- Class and Object
- Implementing a class using UML
- Method overriding and overloading

Create class: use **class** keyword

modifier class ClassName
{
  //attributes and behaviors
}
	public class Vehicle
	{
	  //
	}

Create Object or Instance of a class: use **new** keyword

```
ClassName objectName = new ClassName();
```

Access Modifier:

private: accessible only in this class
protected: accessible only in this package and in all subclasses of this class
public: accessible everywhere, this class is available
default: accessible only in this package and no modifiers are needed

Constructors: Constructors are a special kind of method. They have three peculiarities-

I. A constructor must have the same name as the class itself
II. Constructors do not have a ***return*** type, not even ***void***
III. Constructors are invoked using the ***new*** operator when an object is created. Constructors play the role of initializing objects.

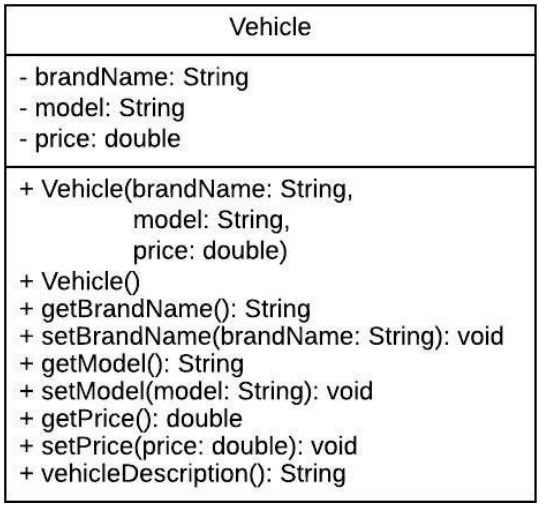
```
public class Vehicle
{
    public Vehicle(String brandName,String model, int price)
  {
        this.brandName = brandName;
        this.model = model;
        this.price = price;
  }
}
```

UML Class Diagram:

A class diagram in the **Unified Modeling Language (UML)** is a type of static structure diagram that describes the structure of a class and their attributes, operations etc. So the basic structure of the UML diagram for a class shown below,

ClassName	*vis* = visibility(+ for public, - for private etc)
vis attribute: type	*attribute* = properties or fields *operation* = method (or constructor)
vis operation(para: type,.....): return type	*para* = parameter name

Now, UML class diagram of the above Vehicle class is shown below,



Vehicle.java

```
package oop;

public class Vehicle {
   private String brandName;
   private String model;
   private int price;

   public Vehicle() {
         this.brandName = "Tata";
         this.model = "Nano 2018";
         this.price = 3000;
   }

   public Vehicle(String brandName,String model, int price) {
         this.brandName = brandName;
         this.model = model;
         this.price = price;
   }

   public String getBrandName() {
         return this.brandName;
   }

   public void setBrandName(String brandName) {
         this.brandName = brandName;
   }

   public String getModel() {
         return this.model;
   }

   public void setModel(String model) {
         this.model = model;
   }

   public int getPrice() {
         return this.price;
   }

   public void setPrice(int price) {
         this.price = price;
   }
```

```
    public String vehicleDescription() {
        String des = "Brief description of your car given below:\n"
                    +"Brand Name: " + this.brandName + "\n"
                     +"Model: " + this.model + "\n"
                    +"Price: " + this.price + " US Dollar";
        return des;
    }
}
```

VehicleMain.java

```
package oop;

public class VehicleMain {

    public static void main(String[] args) {

        Vehicle audi = new Vehicle("Audi", "R8 2018", 177100);
        System.out.println(audi.vehicleDescription()+"\n");

        Vehicle tata = new Vehicle();
        System.out.println(tata.vehicleDescription()+"\n");

        audi.setModel("R8 2019");

        tata.setPrice(2500);

        System.out.println("---After upgrading the data---\n");
        System.out.println(audi.vehicleDescription()+"\n");
        System.out.println(tata.vehicleDescription());
        }
}
```

Task:

Implement a class, name it as ***Circle*** whose UML class diagram is shown below. Make two circle objects using a constructor which has a radius (r) as 1 & 5 respectively and find the diameter (2*r), perimeter (2*π*r) and area ($\pi*r^2$) of these circle objects in a different class, named as ***CircleMain***.