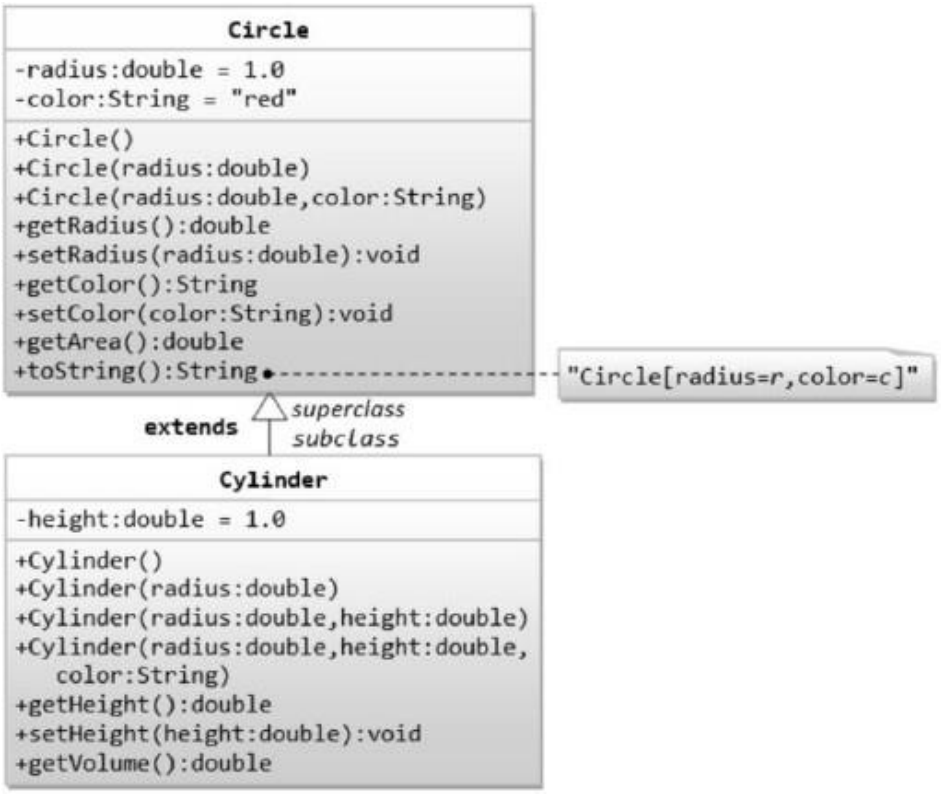
Department of Statistics & Computer Science
University of Kelaniya
Academic Year – 2022/2023
COSC 12043 / BECS 12243 - Object Oriented Programming
Tutorial 12

---

1. A subclass called **Cylinder** is derived from the superclass **Circle** as shown in the below diagram. The Cylinder class overrides the toString() method to return a string representation of the cylinder in the format: "Cylinder[radius=r, height=h, color=c]", where r, h, and c are the values of radius, height, and color, respectively.



   a. Implement the above classes based on the given details.
   b. Write a testDriver class to create below instances and call defined methods for those objects:

```
Circle c1 = new Circle();
Circle c2 = new Cylinder();
Cylinder c3 = new Cylinder (3.1);
Circle c4 = new Circle(2.3);
Circle c5 = new Cylinder (2.3, 4.0);
Cylinder c6 = new Cylinder (3.1, 5.2, "red");
Circle c7 = new Circle(2.1, "yellow");
```

What objects could not call all methods? Give reasons for that.

c. In each class, there is more than one constructor defined. This is called as …… (Fill the blank).

2. Declare the following classes **Manager** and **Clerk** that extends **Employee** Class. The Employee class is given as follows:

```
public abstract class Employee
{
      public String name;
      public double age;
      public double hourRate;
      public abstract double salary(double hours);
      //calculate salary based on hourRate
      public String toString() {
          return "name = "+name+"age = "+age+"hourRate = "+hourRate;
      }
}
```

a. Define all the needed methods and constructors in each of these classes.

b. HR division wants to calculate salary for all employees in this company. Write a Test class for HR division and create objects for the following employees:

   i. Manager Perera is 54 years old and his hourRate is 1000.

   ii. Clerk Saman is 32 years and his hourRate is 600.

   iii. Clerk Nimal is 41 years and his hourRate is 700.

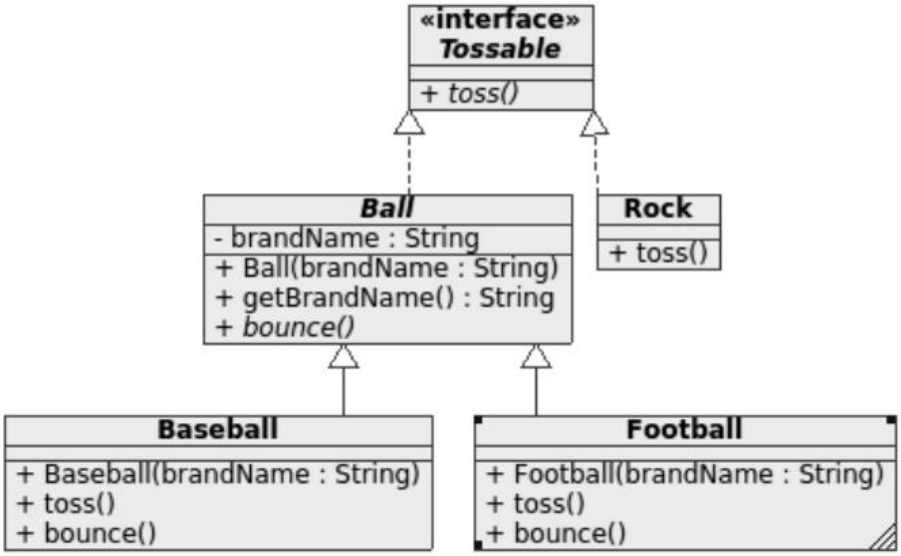
   iv. Clerk Kalani is 28 years and her hourRate is 800.

c. After one week (8 working hours per day), calculate the total payment of above all employees in this company.

d. Can you create Employee objects directly? Why? Employee e1 = new Employee();

e. If you change access modifier public to private in name, age and hourRate instance variables what will happen? Justify your answer.

3. Consider following class hierarchy.



a. Fill in the following table with one of three values.

✓ An object of the indicated type could be stored in a variable of the indicated type.

X An object of the indicated type could not be stored in a variable of the indicated type.

— It is not possible to instantiate an object of the indicated type.

		Variable Type				
		Tossable	Ball	Rock	Baseball	Football
Object Type	Tossable					
	Ball					
	Rock					
	Baseball					
	Football					

b. Which of the following will compile (mark with a C)? Which of the following will execute successfully (mark with an E)?

i. Ball ball = new Football("spalding");

ii. Ball ball = new Football("Spalding"); Baseball baseball (Baseball)ball;

iii. Object obj = new Baseball("spalding");

iv. Object obj = new Baseball("spalding"); Tossable tossable = obj;

v. Tossable tossable = new Baseball("spalding"); Object objtossable;

vi. Rock rock new Baseball("spalding");

vii. Tossable tossable = new Rock();

4. Use the code below to answer the following questions.

```
interface IVehicle {
      // indicate how much a basic tune-up costs
      public double tuneUpCost();

      // determine whether vehicle can hold given num of passengers
      public booleancanCarry (int numPassengers);
}

class Car implements IVehicle {
      int mileage;
      int year;
      int numDoors;

      // constructor goes here

      // indicate whether car was built before given year
      boolean builtBefore(int otherYear) {
            return this.year < otherYear;
      }
}

class Bicycle implements IVehicle {
      int mileage:
      int numGears:

      // constructor goes here
}
```

a. In order to compile above code, what methods you need to add Car and Bicycle classes (except the missing constructors)?

b. Should builtBefore be added to the IVehicle interface? Why or why not?

c. In the Examples class, you want to define a Bicycle as follows:

_____ new KidsBike = new Bicycle(0, 1);

Which types can you use in the blank line (without causing a compile error just with this single line)?

d. Assume you defined a Car object as follows:

IVehicle oldCar = new Car(200000, 1995, 2);

Which methods can you call on oldCar, given the type that you gave it?

e. Assume you defined a Car object as follows:

Car oldCar = new Car (200000, 1995, 2);

Which methods can you call on oldCar, given the type that you gave it?

5. Consider a banking system: each Account contained a Customer and each Customer contained his Account. The classes are given below.

```
class Customer {
      String name;
      int password;
      Account account;

      Customer(String name, Account account) {
            this.name = name;
            this.account = account;
            this.password = genPassword();
      }
 }

class Account {
      int number;
      Customer owner;
      double balance;

      Account(int number, Customer owner) {
            this.number = number;
            this.owner = owner;
            this.balance = 0;
      }
}
```

   a. Someone named Kamal wants to register as a new customer and simultaneously open a new account. Write code to do this using calls to the constructors provided. If you need code beyond calls to the constructors, (i) explain why, and (ii) show the code.

   b. Someone who works in the bank got access to the Customer class in their banking system. He is trying to get the password of the customer called Kamal. Modify the code and write the extra code to prevent the above circumstance.

   c. Anyone who has access to an Account object can read or change the balance. Modify the code and write the extra code to protect balance in Account class.

6. We want to create shapes, which can be either circles (with a radius and a color) or rectangles (with a length, width, and color). Also, we need to create signs (to post around the campus), each of which has a shape (for the background of the sign) and the text (a String) to put on the sign.

   Create classes and interfaces for circles, rectangles, shapes, and signs.

   In order to make signs, we need to make shapes that are large enough to fit the text for the sign. Write a method on shapes fitsText that takes the text as an argument and determines whether the length of the text is shorter than the length/radius of the shape. Call the method length on the string to get the length of a String.

   a. Write the code for the above classes and interfaces as per the description above.
   b. Write a client named ShapeClient and create two circle objects with radius 7 (circle1) and 12 (circle2) whose colors are red and blue respectively and one green rectangle (rectangle1) with length 28 and width 12 in the ShapeClient.
   c. Below three texts are to be put on above shapes to create three sign objects.

      circle1 – “Be silent”

      circle2 – “Study Rooms”

      rectangle1 – “Just 3 min walk for a snack” (should be printed in one line) Create three signs as above and check whether the shapes are large enough to fit the texts.