

# North South University

Department of Electrical and Computer Engineering

CSE 215L: Programming Language II Lab

Lab - 11: Polymorphism

---

## Learning Objectives:

- to learn about polymorphism and its applications

Polymorphism is considered one of the important features of Object-Oriented Programming. The word "poly" means many and "morphs" means forms, So it means many forms. Polymorphism allows us to perform a single action in different ways.

In Java polymorphism is mainly divided into two types:

a) Compile-time Polymorphism: It is also known as static polymorphism. This type of polymorphism is achieved by function overloading or operator overloading. But Java doesn't support operator overloading.

b) Runtime Polymorphism: It is also known as Dynamic Method Dispatch. It is a process in which a call to the overridden method is resolved at runtime. This type of polymorphism is achieved by Method Overriding. Method overriding occurs when a derived class has an updated definition for one of the methods of the base class.

### Ex-1: Compile-time Polymorphism

```
public class Demo {
    // Method 1
    // with 2 integer parameters
    public static int sum(int a, int b) {
        // Returns sum of integer numbers
        return a + b;
    }

    // Method 2
    // with same name but with 2 double parameters
    public static double sum(double a, double b) {
        // Returns sum of double numbers
        return a + b;
    }

    // Method 3
    // with same name but with 3 double parameters
    public static double sum(double a, double b, double c) {
        // Returns sum of double numbers
        return a + b + c;
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        // Calling method by passing
        // input as in arguments
        System.out.println(Demo.sum(3, 5));
        System.out.println(Demo.sum(5.7, 4.2));
        System.out.println(Demo.sum(5.7, 4.2, 1.7));
    }
}
```

#### Ex-2: Runtime Polymorphism:

```
public class Animal {
    public void sound() {
        System.out.println("an animal sounds...");
    }
}

public class Cat extends Animal {
    @Override
    public void sound() {
        System.out.println("a cat sounds: mew...");
    }
}

public class Dog extends Animal {
    @Override
    public void sound() {
        System.out.println("a dog sounds: barking...");
    }
}

public class Main {
    public static void main(String[] args) {
        // sound for an animal
        Animal a1 = new Animal();
        a1.sound();

        // sound for a cat
        Animal a2 = new Cat();
        a2.sound();

        // sound for a dog
        Animal a3 = new Dog();
        a3.sound();
    }
}
```

**Lab Task:**

Circle
- radius: double
+ Circle() + Circle(radius: double) + setRadius(radius: double): void + getRadius(): double + getArea(): double + toString(): String

Cylinder
- height: double
+ Cylinder() + Cylinder(radius: double, height: double) + setHeight(height: double): void + getHeight(): double + getArea(): double + getVolume(): double + toString(): String

Suppose, Circle is a base class and Cylinder is a derived class. Now accomplish the following tasks using your polymorphism knowledge in java.

a) Create an array of 3 Cylinder instances and you must read their radius and height from the user.

b) Print the status of all Cylinder instances. [Hint: toString() method format for printing in Circle: (radius: p, area: q) and Cylinder: (height: p, radius: q, area: r, volume: s); where p, q, r, s are some decimal values]

c) Print the sum of the height, radius, area, volume of all cylinders and show the result in a table format given below.

no	height	radius	area	volume
1	p1	q1	r1	s1
2	p2	q2	r2	s2
3	p3	q3	r3	s3
Total	p	q	r	s