CSE 341

Class -03
12/01/2025

Constructors

1. Default Constructor → when no constructor is declared
   ↳ default is not always executed
2. Parameterized Constructor
3. Copy Constructor

Polymorphism > Method Overloading

Many Forms

Overloading Identifiers

↳ Number of parameters and/or

↳ parameter types <type signature>

↳ return type different হলে uniquely identify করা possible না

```
class sum{
  int add(int a, intb){
       return a+b;
  }
  double add(double a, double b){
       return a+b;
  }
  int add(int a, intb, int c){
       return a+b+c;
  }
}

public class Main{
 public static void main(String[]args)
 { sum s = new sum();
   System.out.println(
       s.add(1,2) +
       s.add(2.5,1.5)+
       s.add(1,2,3)
   );
 }
}
```

Take Object as Parameter

```
class add{
   □ int a,b;
   add(int a, int b){
       this.a = a;
       this.b = b;
   }

   boolean get(add ob){
      if(Ob.a==a && ob.b==b) return true;
      return false;
   }
}

public class Main{
   public static void Main( String[] args){
      add ob1= new add(5,6);
      add ob2 = new add(10,100);
      add ob3 = new add();
         will cause error
      System.out.println(ob1.get(ob2));
   }
}
```

access modifier default

{ this pointer }

দুইটা object same memory কে point করছে।

Same variable হলে "this" pointer use করা উচিত

↳ add(int a, int b){
this.a = a;
this.b = b; }

#Returning Object

```
add ~~get~~ back(){...}

add ~~get~~ back(add ob){
    add temp = new add(3,4);
    temp.a = a+ob.a;
    temp.b = b+ob.b;
    return temp;
}
```

① declare a method that take an object as parameter

② declare a method that returns an object

```
public class Main{
 public static void main(Str[] args){
  add ob3=ob2.back(ob2);
  System.out.
     println(ob3.a+" "
           +ob3.b);
 }
}
```

```
class Rectangle{
    double area;

    Rectangle(double area){
        this. area = area;
    }

    Rectangle get(Rectangle r1, Rectangle r2){
        Rectangle temp= new Rectangle (1);
        temp. area = r1.area * r2.area;
        return temp;
    }
}

public class Main{
    public static void main(string[] args){
        Rectangle rX = new Rectangle (1);
        Rectangle r1 = new Rectangle (15);
        Rectangle r2 = new Rectangle (2);
        ~~System.out.println(rX(r1, r2))~~;
        Rectangle r3 = rX.get(r1,r2);
    }
}       System.out.println(r3.area);
```

Array of Objects

```
class Student{
        int id;
        String name;

    Student(int id, String name){
        this.id = id;
        this.name = name;
    }

    void show(){
        System.out.println(
          "id:" +id +
          "name: " +name
        );
    }
}
public class Main{
  public static void main( String[] args){
    Student s[] = new student [3];
    ↑type   ↑     ↑         ↑       ↑ no. of
           array  new       type      Objects
           name   Keyword

  <1টা classর 3টা instance or object>

  for(int i=0; i<s.length ; i++){
      S[i].student(123, "Rahim");
  }    ↳ S[i] = new Student (123, "Rahim");
```

```
for(student s1: s){
        s1.show();
}


Student(){ id=45;
          name="Karim";}
```

```
Student  s[] = {
        new  Student(),
        new  Student(123, "Rahim");
        new  Student (13, "ABC");
    };


for (student s1:s) {
        s1.show();
}
```