```
// Class And Object


public class Main
{
	public static void main(String[] args) {

		Programming p=new Programming();
		p.id=1;
		p.name="Anil";
		p.course="Java Programming";

		System.out.println("Id :- "+p.id+"  Name :- "+p.name+"  Course :- "+p.course);

		Programming q=new Programming();
		q.id=2;
		q.name="Sunil";
		q.course="Advance Java Programming";

		System.out.println("Id :- "+q.id+"  Name :- "+q.name+"  Course :- "+q.course);
	}
}

class Programming{

   int id;
   String name;
   String course;

}
```

```
// Encapsulation


public class Main
{
	public static void main(String[] args) {


		Programming p=new Programming();
		p.setId(1);
		p.setName("Anil");
		p.setCourse("Java Programming");


		System.out.println("Id :- "+p.getId()+"  Name :- "+p.getName()+"  Course :- "+p.getCourse());


		Programming q=new Programming();
		q.setId(2);
		q.setName("Sunil");
		q.setCourse("Advance Java Programming");


		System.out.println("Id :- "+q.getId()+"  Name :- "+q.getName()+"  Course :- "+q.getCourse());
	}
}

class Programming{


   private int id;
   private String name;
   private String course;


   public void setId(int id){
      this.id=id;
```

```
    }

    public void setName(String name){
       this.name=name;
    }

    public void setCourse(String course){
       this.course=course;
    }

    public int getId(){
       return this.id;
    }

    public String getName(){
       return this.name;
    }

    public String getCourse(){
       return this.course;
    }

}
```

```
// Single Inheritence

public class Main
{
	public static void main(String[] args) {

		A a=new A();
		B b=new B();

		a.display();
		b.display();

	}
}

class A{

   public void display(){
      System.out.println("Class A");
   }

}

class B extends A{

   public void display(){
      System.out.println("Class B");
   }

}
```

```
// Multilevel Inheritence


public class Main
{
	public static void main(String[] args) {

		A a=new A();
		B b=new B();
		C c=new C();

		a.display();
		b.display();
		c.display();

	}
}

class A{

   public void display(){
      System.out.println("Class A");
   }

}

class B extends A{

   public void display(){
      System.out.println("Class B");
   }
```

```
}

class C extends B{

    public void display(){
        System.out.println("Class C");
    }

}
```

```
// Hierarchical Inheritence


public class Main
{
	public static void main(String[] args) {

		A a=new A();
		B b=new B();
		C c=new C();

		a.display();
		b.display();
		c.display();

	}
}

class A{

   public void display(){
      System.out.println("Class A");
   }

}

class B extends A{

   public void display(){
      System.out.println("Class B");
   }
```

```
}

class C extends A{

    public void display(){
        System.out.println("Class C");
    }

}
```

```
// Runtime Polymorphism

public class Main
{
	public static void main(String[] args) {

		A a=new A();
		A b=new B();
		A c=new C();

		a.display();
		b.display();
		c.display();

	}
}

class A{

   public void display(){
      System.out.println("Class A");
   }

}

class B extends A{

   public void display(){
      System.out.println("Class B");
   }
```

```
}

class C extends B{

    public void display(){
        System.out.println("Class C");
    }

}
```

```
// Compile Time Polymorphism

public class Main
{
	public static void main(String[] args) {

		A a=new A();

		a.add(5,5);
		a.add(5,5,5);
		a.add(5,5,5,5);

	}
}

class A{

   public void add(int a,int b){
      System.out.println("Addition of a and b :- "+(a+b));
   }

   public void add(int a,int b,int c){
      System.out.println("Addition of a, b and c :- "+(a+b+c));
   }

   public void add(int a,int b,int c,int d){
      System.out.println("Addition of a, b, c and d :- "+(a+b+c+d));
   }

}
```

```
// Method Overriding


public class Main
{
	public static void main(String[] args) {

		A a=new A();
		B b=new B();
		C c=new C();

		a.display();
		b.display();
		c.display();

	}
}

class A{

   public void display(){
      System.out.println("Class A");
   }

}

class B extends A{

   public void display(){
      System.out.println("Class B");
   }
```

```
}

class C extends B{

    public void display(){
        System.out.println("Class C");
    }

}
```

```
// Method Overloading


public class Main
{
	public static void main(String[] args) {

		A a=new A();

		a.add(5,5);
		a.add(5,5,5);
		a.add(5,5,5,5);

	}
}

class A{

   public void add(int a,int b){
      System.out.println("Addition of a and b :- "+(a+b));
   }

   public void add(int a,int b,int c){
      System.out.println("Addition of a, b and c :- "+(a+b+c));
   }

   public void add(int a,int b,int c,int d){
      System.out.println("Addition of a, b, c and d :- "+(a+b+c+d));
   }

}
```

```
// Abstraction

// Abstract Class

public class Main
{
        public static void main(String[] args) {

            Samsung s=new Samsung();
            s.company();
            s.model();
            s.RAM();

        }
}

abstract class Mobile{

   abstract public void company();

   abstract public void model();

   abstract public void RAM();
}

class Samsung extends Mobile{

   public void company(){
      System.out.println("Samsung");
   }
```

```
    public void model(){
        System.out.println("MS1");
    }

    public void RAM(){
        System.out.println("4GB");
    }

}
```

```
// Abstraction

// Interface

public class Main
{
        public static void main(String[] args) {

            Samsung s=new Samsung();
            s.company();
            s.model();
            s.RAM();

        }
}

interface Mobile{

   abstract public void company();

   abstract public void model();

   abstract public void RAM();
}

class Samsung implements Mobile{

   public void company(){
      System.out.println("Samsung");
   }
```

```
    public void model(){
        System.out.println("MS1");
    }

    public void RAM(){
        System.out.println("4GB");
    }

}
```

```
// Constructor And Constructor Overloading


public class Main
{
        public static void main(String[] args) {


           Add a=new Add(5,5);
                Add b=new Add(5,5,5);
                Add c=new Add(5,5,5,5);


        }
}


class Add{


   Add(int a,int b){
      System.out.println("Addition of a and b :- "+(a+b));
   }


  Add(int a,int b,int c){
      System.out.println("Addition of a, b and c :- "+(a+b+c));
   }


   Add(int a,int b,int c,int d){
      System.out.println("Addition of a, b, c and d :- "+(a+b+c+d));
   }


}
```