Practical 04

```
01. public class Employee
    {
        private int empID;
        private String empName,empDesignation;

        public void setEmpID(int empID)
        {
                this.empID=empID;
        }
        public void setEmpName(String empName)
        {
                this.empName=empName;
        }
        public void setEmpDesignation(String empDesignation)
        {
                this.empDesignation=empDesignation;
        }
        public int getEmpID()
        {
                return empID;
        }
        public String getEmpName()
        {
                return empName;
        }
```

```
        public String getEmpDesignation()
        {
                return empDesignation;
        }
    }
public class Test1
{
        public static void main(String[] args)
        {
                Employee bogdan=new Employee();
                bogdan.setEmpID(001);
                bogdan.setEmpName("Bogdan");
                bogdan.setEmpDesignation("Lecturer");
                System.out.println("Employee ID: "+bogdan.getEmpID());
                System.out.println("Employee Name: "+bogdan.getEmpName());
                System.out.println("Employee Designation: "+bogdan.getEmpDesignation()+"\n");

                Employee bird=new Employee();
                bird.setEmpID(002);
                bird.setEmpName("Bird");
                bird.setEmpDesignation("Lecturer");
                System.out.println("Employee ID: "+bird.getEmpID());
                System.out.println("Employee Name: "+bird.getEmpName());
                System.out.println("Employee Designation: "+bird.getEmpDesignation());
      }
}
```

```
02. class SuperB
    {
        int x;
        void setIt (int n)
        {
                x=n;
        }
        void increase ()
        {
                x=x+1;
        }
        void triple ()
        {
                x=x*3;
        }
        int returnIt ()
        {
                return x;
        }
    }
class SubC extends SuperB
{
        void triple () {x=x+3;} // override existing method
        void quadruple () {x=x*4;} // new method
}
```

```
public class TestInheritance
{
	public static void main(String[] args)
	{
		SuperB b = new SuperB();
		b.setIt(2);
		b.increase();
		b.triple();
		System.out.println( b.returnIt() );
		SubC c = new SubC();
		c.setIt(2);
		c.increase();
		c.triple();
		System.out.println( c.returnIt() );
	}
}
```

Result – 9

6

03.

```
public class Person
    {
	private String name;
	private int id;
	public void setName(String name)
	{
		this.name=name;
	}
```

```
        public void setId(int id)
        {
                this.id=id;
        }
        public String getName()
        {
                return name;
        }
        public int getId()
        {
                return id;
        }
    }
public class Student extends Person
{
        private String course;
        public void setCourse(String course)
        {
                this.course=course;
        }
        public String getCourse()
        {
                return course;
        }
}
```

```
public class Lecturer extends Person
{
	private String programme;
	public void setProgramme(String programme)
	{
		this.programme=programme;
	 }
	public String getProgramme()
	{
		return programme;
	}
}
public class Test
{
	public static void main(String[] args)
	{
		Student s1=new Student();
		s1.setName("Anne");
		s1.setId(0001);
		s1.setCourse("Networking");
		System.out.println("Student Name: "+s1.getName());
		System.out.println("Student ID: "+s1.getId());
		System.out.println("Student Course: "+s1.getCourse()+"\n");
```

```
            Lecturer l1=new Lecturer();
            l1.setName("Mr.Perera");
            l1.setId(001);
            l1.setProgramme("Networking and Security");
            System.out.println("Lecturer Name: "+l1.getName());
            System.out.println("Lecturer ID: "+l1.getId());
            System.out.println("Lecturer Programme: "+l1.getProgramme());
        }
}
04. public class Animal
    {
    }
    public class Mammal extends Animal
   {
   }
   public class Reptile extends Animal
   {
   }
   public class Dog extends Mammal
   {
```

```
    public static void main(String args[])
    {
            Animal a = new Animal();
            Mammal m = new Mammal();
            Dog d = new Dog();
            System.out.println(a instanceof Animal);
            System.out.println(m instanceof Mammal);
            System.out.println(d instanceof Animal);
    }
}
```