Practical 4

1)

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


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


    public void setEmpDesignation(String empDesignation) {
        this.empDesignation = empDesignation;
    }
}
```

Test class

```
public class TestEmployee {
    public static void main(String[] args) {
        Employee bogdan = new Employee();
        bogdan.setEmpID(1);
        bogdan.setEmpName("Mr. Bogdan");
        bogdan.setEmpDesignation("Manager");

        Employee bird = new Employee();
        bird.setEmpID(2);
        bird.setEmpName("Ms. Bird");
        bird.setEmpDesignation("Developer");

        System.out.println("Employee 1:");
        System.out.println("ID: " + bogdan.getEmpID());
        System.out.println("Name: " + bogdan.getEmpName());
        System.out.println("Designation: " + bogdan.getEmpDesignation());

        System.out.println("\nEmployee 2:");
        System.out.println("ID: " + bird.getEmpID());
        System.out.println("Name: " + bird.getEmpName());
        System.out.println("Designation: " + bird.getEmpDesignation());
    }
}
```

2)

```
  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; // overriding 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());
    }
}
```

3)

```
class Person {
  private String name;
  private int id;

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

  public String getName() {
    return name;
  }

  public void setID(int id) {
    this.id = id;
  }

  public int getID() {
    return id;
  }
}
```

```
}



class Student extends Person {
    private String course;

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

    public String getCourse() {
        return course;
    }
}





class Lecturer extends Person {
    private String programme;

    public void setProg(String programme) {
        this.programme = programme;
    }

    public String getProg() {
        return programme;
    }
}
```

```
public class TestPerson {
    public static void main(String[] args) {
        Student student = new Student();
        student.setName("John Doe");
        student.setID(1001);
        student.setCourse("Computer Science");

        Lecturer lecturer = new Lecturer();
        lecturer.setName("Jane Smith");
        lecturer.setID(2001);
        lecturer.setProg("Software Engineering");

        System.out.println("Student:");
        System.out.println("Name: " + student.getName());
        System.out.println("ID: " + student.getID());
        System.out.println("Course: " + student.getCourse());

        System.out.println("\nLecturer:");
        System.out.println("Name: " + lecturer.getName());
        System.out.println("ID: " + lecturer.getID());
        System.out.println("Programme: " + lecturer.getProg());
    }
}
```

4)

```
public class Animal {

   // Code for the Animal class

}


public class Mammal extends Animal {

   // Code for the Mammal class

}


public class Reptile extends Animal {

   // Code for the Reptile class

}
```