Java Assignment

192211633
T. Rajesh

1. Simple Inheritance :-

```
class animal {
    string name;
    void makesound() {
        system.out.println("Animal makes a sound");
    }
}
class Dog extends Animal {
    @Override
    void makesound() {
        system.out.println("The dog barks");
    }
}
Public class main{
    Public static void main(String[] args){
        Dog dog = new Dog();
        dog.makesound();
    }
}
```

2. Constructer Inheritance

```
class Person{
    String name;
    int age;
    Person(string name, int age){
        this.name = name;
        this.age = age;
    }
}
```

```
class student extends person { string grade;
 student (string name, int age, string
grade ) {
        super (name, age);
        this.grade = grade;
    }
      void display() {
    student student = new student
("John", 20, "A");
          student.display();
    }
}
```

3. Multilevel Inheritance :-

```
class vehicle {
    int speed;
    String fuelType;
  vehicle (int speed, string fuelType) {
    This.speed = speed;
     this.fuelType = fultype;
      }
  }
class car extends vehicle {
     int numberofDoors;
  car (int speed, string fuelType,
int numberofDoors ) {
      super (speed, fuelType);
```

```
}
}
  Public Main {
      Public static void main (string[]
args) {
      Electriccar electriccar = new
ElectricCar (120, "Eletric", 4, 85);
electric:displayproperties ();
    }
  }
```

Method Overriding in Inheritance

```
class shape {
    void draw() {
System.out.println ("Drowing a shape");
  }
}
class circle extends shape{
    @override
    void draw() {
System.out.println("Drawing a circle");
  }
}
 Public class main {
 Public static void main(string[]args){
      shape circle = new circle();
```

```
    circle.draw();
    }
}
```

5. Inheritance and Access Modifiers :-

```
class Employee {
    Private string privateField =
"Private";
    Protected string
Protected Field = "Protected";
    Public string public field = "Public";
    }
class Manager extends Employee
  { void displayfields() {
System.out.println(private field);
}
}
  Public class main {
      Public static void main
(string[] args) {
          Manager manager = new
Manager();
manager.displayfields();
    }
}
```