CSA0914 - PROGRAMMING IN JAVA FOR RASPBERRY PI

ASSIGNMENT - 4

T.R. Tharun

192211541

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 bark");
    }
}
Public class main{
        Public static void main (string[]
        args){
        Dog dog = new Dog ();
        dog . makesound ();
    }
}
```

2. construiter Inheritance

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

```
class student extends person { string grad;
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 fuel type
  {
      This.speed = speed;
      this.fuel type = fuel type;
    }
  }
class car extends vehicle {
      int number of doors;
car (int speed, string fuel type,
 int number of doors) {
      super(speed, fuel type);
```

```
}
}
Public Main {
    Public static void main (string
[] args) {
        Electric car electriccar = new
electric.displayproperties();
    }
}
```

4. Method overiding in Inheritance

```
class shape {
    void draw() {
system.out.println("Drowing a
shape") }
}
class circle extends shape {
    @override
    void draw() {
system.out.println("Drowing 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 =
"Poivate";
    Protected string
Protected field = "Protected";
Public string public field = "Public";
    }
class manager extends employee
    {
    void displayfield() {
System.out.println(privatefield);
    }
}
    public class main {
    public static void main(string
[] args) {
    manager.manager = new
manager();
    manager.displayfields();
    }
}
```