JAVA PROGRAMMING FOR RASPBERRY PI - CSA0914

ASSIGNMENT - 4

P. Durga
192211195

Simple Inheritance

```
class Animal {
    string name;
    void makeSound(){
System.out.print("Anima) makes a sound");
}}
  class Dog extends Animal {
    @Override
    void makesound(){
   System.out.print("The Dog bark");
  }}
    Public class main {
      Public static void main (string()args){
     Dog dog= new Dog();
     dog.make sound(); }}
```

Constructer Inheritance

```
class Person {
        string name;
        int age;
        Person (string name int age){
```

Private

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

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 numberofDoors;
    car (int speed, string fuel type,
```

```
    int numberofDoors){
        super(speed, fuelType); }}

Public main{
    Public static void main(String[] args){
    electricCar electricCar = new electric.display
    Properties(); }}
```

4) Method overriding in Inheritance

```
class shape{
    Void draw(){
    System.out.print("Drawing a shape")}}
class circle extends shape{
    @override
    void draw(){
System.out.print("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 display field() {
System.out.print(Private field);
}
}

    Public class main {
    Public static void main(string [] args) {
Manager . manager = new
    manager();
    manager. display field s[];
}
}
```