

ASSIGNMENT

≈04

NAME:- T. Parvan Kumar

REG NO:- 192210290

COURSE CODE:- CSA0914

COURSE NAME:- Programming in Java for
Raspberry Pi

Submitted To:-

Dr. Hemavathi. R

1. Inheritance Pseudocode :-

PROCEDURE Inheritance

Animal

name: string

method makesound()

Print "The animal makes a sound"

end method

end Animal

Dog inherits from Animal

override method makesound()

Print "The dog barks"

end method

end class

Main

my dog : Dog // declare

Call my dog.makesound.

End Program.

Program :-

Class Animal {

String name;

Public void makesound () {

System.out.println ("Animal make sound");

}

}

class Dog extends Animal {

Public void makesound () {

System.out.println ("Dog barks");

}

}

Public class Main {

Public Static void main (String args) {

```
Animal animal = new Animal ();
Animal.name = "Generic Animal";
animal.makeSound();
```

```
Dog dog = new Dog();
dog.name = "Buddy";
dog.makeSound();
```

```
}
}
```

2) Constructors inheritance :-

Program:-

```
Class Person {
    String name;
    int age;
```

```
    Public Person (String name, int age) {
        this.name = name;
        this.age = age;
```

```
    Public void displayInfo () {
        System.out.println ("Name: " + name + ", Age: " + age);
```

```
    }
}
class Student extends Person {
```

```
    String grade;
```

```
    Public Student (String name, int age) {
```

```
        Super (name, age);
```

```

        this.grade = grade;
    }
    public void displayInfo () {
        System.out.println ("grade: " + grade);
    }
}
public class Main () {
    public static void main (String [] args) {
        Student student = new Student ("Alice", 20, "A");
        student.displayInfo ();
    }
}

```

3) Multilevel inheritance:

Program:

```

class vehicle {
    String brand;
    public vehicle (String brand) {
        this.brand = brand;
    }
    public void displayInfo () {
        System.out.println ("Brand: " + brand);
    }
}
class car extends vehicle {
    String model;
    public car (String brand, String model) {
        super (brand);
        this.model = model;
    }
}

```

```
Public void displayInfo() {
```

```
super.displayInfo();
```

```
System.out.println("Battery Capacity: " + batteryCapacity +  
"kWh");
```

```
}
```

```
}
```

```
Public static void main (String[] args) {
```

```
ElectricCar car = new ElectricCar ("Tesla", "Model S", 100);
```

```
}
```

```
}
```

4) Method overriding in inheritance:-

Program:-

```
Class shape {
```

```
Public void draw() {
```

```
System.out.println("Drawing a shape");
```

```
}
```

```
}
```

```
Class Circle extends shape {
```

```
Public void draw() {
```

```
System.out.println("Drawing a circle");
```

```
}
```

```
}
```

```
Public class Main {
```

```
Public static void main (String[] args) {
```

```
Shape shape = new Shape();
```

```
shape.draw();
```

```
}
```



```
Circle circle = new Circle (1);
```

```
circle.draw (1);
```

```
Rectangle rectangle = new Rectangle (1);
```

```
Rectangle.draw (1);
```

```
}
```

```
}
```

5) Inheritance and Access Modifiers:-

Program:-

```
Class Employee {
```

```
    Private int id;
```

```
    Protected String name;
```

```
    Public double salary;
```

```
    Public Employee (int id, String name, double salary) {
```

```
        this.id = id;
```

```
        this.name = name;
```

```
        this.salary = salary;
```

```
    }
```

```
    Public void display ManagementInfo () {
```

```
        System.out.println ("ID: " + id);
```

```
        System.out.println ("Name: " + name);
```

```
        System.out.println ("Salary: " + salary);
```

```
    }
```

```
    Public class Main {
```

```
        Public static void main (String [] args) {
```

```
            Manager Manager = new Manager (1, "Alice", 75000.0);
```

```
            Manager.display ManagementInfo ();
```

```
            Manager.display EmployeeInfo ();
```

```
        }
```

```
    }
```