

(16) break an integer & sequence  
of digits

## ASSIGNMENT-4

Name:- ch.Mahanvitha

Reg.No:- 192211785

Course Code:- CSA0914

Course Name:- Programming  
in Java for  
Raspberry Pi

Faculty :- Dr. Hemavathi

1) Java program for simple Inheritance example

Input:- Create an object

Output:- The dog barks

Program:-

```
Class Animal {
```

```
    String name;
```

```
    Public Animal (String name) {
```

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

```
}
```

```
    Public void make sound () {
```

```
        System.out.println ("Animal makes  
sound");
```

```
}
```

```
Class dog extends Animal {
```

```
    Public dog (String name) {
```

```
        Super (name);
```

```
}
```

```
Public class Simple Inheritance
```

```
example {
```

```
Dog dog = new Dog ("Buddy");
```

```
dog.make sound();
```

```
}
```

```
{
```

## 2) Constructors Inheritance

Example:-

Input:- Create a Student object with name, age and grade

Output:- Display the name, age of the Student.

Pseudocode:- Program

class person {

    String name;

    int age;

    public person(String name, int age) {

        this.name = name;

        this.age = age;

}

Student extends person {

    String grade;

    public Student (String name, int age) {

        super(name, age);

        this.grade() = grade;

    public class constructor inheritance {

        public static void main (String[] args) {

            Student student = new Student("lo");

            student.display();

## Multilevel Inheritance

**Input:-** Create electricCar and properties  
are speed, fuelType, batteryCapacity.  
**Output:-** Display all properties of  
electricCar.

Pseudocode:-

Program:-

```
Class vehicle {
```

```
    int speed;
```

```
    String fuelType;
```

```
public vehicle (int speed, String  
fuelType) {
```

```
    this.speed = speed;
```

```
    this.fuelType = fuelType;
```

```
}
```

```
Class car vehicle {
```

```
    Super (speed, fuelType);
```

```
}
```

```
Public class multilevel inheritance {
```

```
    Public static void main (String []
```

```
        args) {
```

```
        electricCar electricCar = new electric  
(electricCar);
```

```
        car
```

```
        electricCar display();
```

```
}
```

#### 4) Method overriding Inheritance

Input:- Create objects of circle, rectangle and their draw  
Output:- Drawing a circle for circle, rectangle for rectangle.

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 MethodOverriding {  
    Shape circle = new Circle();  
    Shape rectangle = new Rectangle();  
    circle.draw();  
    rectangle.draw();  
}
```

Inheritance and ac  
Input:- Create manager  
Output:- Private atm  
Bank Program  
Class

Inheritance  
of circle, rectangle  
Draw  
circle for circle,  
for rectangle.

{ sequence of  
3 4 5 6

Inheritance and access modifiers.

**Input:-** Create Manager object &

**Output:-** private access.

**Program:-**

```
Class employee {  
    private String name;  
    protected String int id;  
    public String department;  
    public employee (String name, int id,  
                    String department);  
    this.name = name;  
    this.id = id;  
    this.dep = dep;  
Class manager extends manager  
employee {  
    public class Access modifiers {  
        public static void main (String []  
                               args) {  
            manager manager = new manager  
            ( );  
            manager.display info ();  
        }  
    }  
}
```