Object Oriented Programming using Java Laboratory (DJS23FLES201)
Academic Year 2023-24

EXPERIMENT NO. 8

NAME – SHAIKH ARSHAD AJIJ

BRANCH : CSE-ICB

ROLL NO : B007

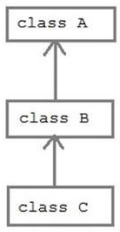
SAP ID : 60019230064

AIM / OBJECTIVE:

To implement Inheritance and super keyword

1. To implement Inheritance and super keyword

a. WAP to demonstrate the role of Constructors in inheritance in the following class diagram.



Code-

```
class A {
    A(int x) {
      System.out.println("This is parent class A with value of x : " + x);
    }
  }

    class B extends A{

      B(int x,int y){
```

```
        super(x);
        System.out.println("This is Child class B of parent A with value of y : " + y);
    }

}

class C extends B{

    C(int x, int y, int z){
        super(x,y);
        System.out.println("This is Child class C of parent B with value of z : " + z);
    }

}

public class Main {
    public static void main(String[] args) {
        C obj = new C(2,3,4);

    }
}
```

Output-

```
C:\Users\Arshad\Desktop\study\java>java Main
This is parent class A with value of x : 2
This is Child class B of parent A with value of y : 3
This is Child class C of parent B with value of z : 4
```

b. WAP to create a super class having a variable. Let the variable be initialized to some value within a constructor. This class should have a method display () to display the initial value of the variable. Derive a sub class that accesses the constructor, variable and method of the super class using super keyword.

Object Oriented Programming using Java Laboratory (DJS23FLES201)
Academic Year 2023-24

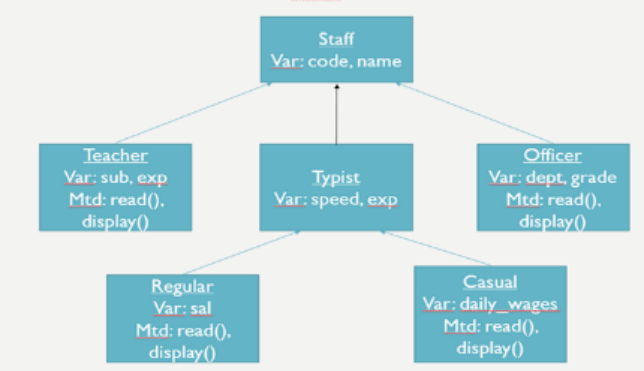
Code-

```
class Superclass {
    int variable;

    Superclass(int x) {
        variable = x;
    }

    void display() {
        System.out.println("Initial value of variable: " + variable);
    }
}


class Subclass extends Superclass {
    Subclass(int x) {
        super(x);
        super.display();
    }

}

public class Main {
    public static void main(String[] args) {
        // Creating an object of Subclass
        Subclass obj = new Subclass(10);
    }
}
```

Output-

```
C:\Users\Arshad\Desktop\study\java>java Main
Initial value of variable: 10
```

c. Display data of the specialized classes given in the following class diagram.



Code-

```
import java.util.*;
public class Main {

   public static void main(String[] args) {
      Scanner sc=new Scanner(System.in);
      Teacher t=new Teacher();
      Officer o=new Officer();
      Regular rt= new Regular();
      Casual ct=new Casual ();
      String teacher_name,teacher_code,subject;
      int experience;
      System.out.println("Enter teacher code: ");
      teacher_code=sc.nextLine();
      System.out.println("Enter techer name: ");
      teacher_name=sc.nextLine();
```

```
System.out.println("Enter teacher's subject: ");
subject=sc.nextLine();
System.out.println("Enter teacher experience: ");
experience=sc.nextInt();
sc.nextLine();
String o_name,o_code,o_dept,o_grade;
System.out.println("\nEnter Officer's code: ");
o_code=sc.nextLine();
System.out.println("Enter Officer's name: ");
o_name=sc.nextLine();
System.out.println("Enter Officer's department: ");
o_dept=sc.nextLine();
System.out.println("Enter Officer's grade: ");
o_grade=sc.nextLine();
String rcode,rname,rspeed;
int rexp,rsal;
System.out.println("\nEnter regular typist's code: ");
rcode=sc.nextLine();
System.out.println("Enter regular typist's name: ");
rname=sc.nextLine();
System.out.println("Enter regular typist's speed: ");
rspeed=sc.nextLine();
System.out.println("Enter regular typist's experience: ");
rexp=sc.nextInt();
System.out.println("Enter ragular typist's salary: ");
rsal=sc.nextInt();
sc.nextLine();
String ccode,cname,cspeed;
int cexp,cwages;
System.out.println("\nEnter casual typist's code: ");
ccode=sc.nextLine();
System.out.println("Enter casual typist's name: ");
cname=sc.nextLine();
```

```
        System.out.println("Enter casual typist's speed: ");
        cspeed=sc.nextLine();
        System.out.println("Enter casual typist's experience: ");
        cexp=sc.nextInt();
        System.out.println("Enter casual typist's salary: ");
        cwages=sc.nextInt();
        t.read(teacher_code,teacher_name,subject, experience);
        o.read(o_code,o_name,o_dept,o_grade);
        rt.read(rcode,rname,rspeed,rexp,rsal);
        ct.read(ccode,cname,cspeed,cexp,cwages);
        t.disp();
        o.disp();
        rt.disp();
        ct.disp();
    }
}
class Staff
{
    String code;
    String name;
}
class Teacher extends Staff
{
    String sub;
    int exp;
    void read(String c,String n,String s,int e)
    {
        code=c;
        name=n;
        sub=s;
        exp=e;
    }
    void disp()
```

```
    {
        System.out.println("\nTeacher's code is: "+ code);
        System.out.println("Teacher's name is: "+ name);
        System.out.println("Teacher's subject is: "+ sub);
        System.out.println("Teacher's experience is: "+ exp+ " years.");


    }
}
class Officer extends Staff
{
    String dept;
    String grade;
    void read(String c,String n,String d,String g)
    {
        code=c;
        name=n;
        dept=d;
        grade=g;
    }
    void disp()
    {
        System.out.println("\nOfficer's code is: "+ code);
        System.out.println("Officer's name is: "+ name);
        System.out.println("Officer's dept is: "+dept);
        System.out.println("Officer's grade is: "+ grade);
    }
}
class Typist extends Staff
{
    String speed;
    int exp;
}
class Regular extends Typist
```

```
{
   int sal;
   void read(String c,String n,String s,int e,int  sl)
   {
      code=c;
      name=n;
      speed=s;
      exp=e;
      sal=sl;
   }
   void disp()
   {
      System.out.println("\nRegular typist's code is: "+ code);
      System.out.println("Regular typist's name is: "+ name);
      System.out.println("Regular typist's speed is: "+speed);
      System.out.println("Regular typist's experience is: "+exp+ " years");
      System.out.println("Regular typist's salary is: "+sal);
   }
}
class Casual extends Typist
{
   int daily_wages;
   void read(String c,String n,String s,int e,int  dw)
   {
      code=c;
      name=n;
      speed=s;
      exp=e;
      daily_wages=dw;
   }
   void disp()
   {
      System.out.println("\nCasual typist's code is: "+ code);
```

```
        System.out.println("Casual typist's name is: "+ name);
        System.out.println("Casual typist's speed is: "+speed);
        System.out.println("Casual typist's experience is: "+exp);
        System.out.println("Casual typist's daily wages are: "+daily_wages);
    }
}
```

Output-

```
C:\Users\Arshad\Desktop\study\java>java Main
Enter teacher code:
123
Enter techer name:
Shruti
Enter teacher's subject:
Maths
Enter teacher experience:
10

Enter Officer's code:
6464
Enter Officer's name:
Chaudhari
Enter Officer's department:
Chemistry
Enter Officer's grade:
A

Enter regular typist's code:
47
Enter regular typist's name:
Krish
Enter regular typist's speed:
72
Enter regular typist's experience:
4
Enter ragular typist's salary:
15000
```

Object Oriented Programming using Java Laboratory (DJS23FLES201)
Academic Year 2023-24

```
Enter regular typist's experience:
4
Enter ragular typist's salary:
15000

Enter casual typist's code:
663
Enter casual typist's name:
Krishna
Enter casual typist's speed:
56
Enter casual typist's experience:
2
Enter casual typist's salary:
10000

Teacher's code is: 123
Teacher's name is: Shruti
Teacher's subject is: Maths
Teacher's experience is: 10 years.

Officer's code is: 6464
Officer's name is: Chaudhari
Officer's dept is: Chemistry
Officer's grade is: A

Regular typist's code is: 47
Regular typist's name is: Krish
Regular typist's speed is: 72
Regular typist's experience is: 4 years
Regular typist's salary is: 15000

Casual typist's code is: 663
Casual typist's name is: Krishna
Casual typist's speed is: 56
Casual typist's experience is: 2
Casual typist's daily wages are: 10000
```

CONCLUSION:

Constructors play a crucial role in inheritance, especially in scenarios involving multiple inheritance. In Java, when a subclass is instantiated, constructors of its superclass(es) are automatically invoked, ensuring proper initialization of inherited fields and behaviour. By using super() keyword in subclass constructors, we can explicitly call superclass constructors, passing necessary arguments to initialize superclass members.

Website References: javapoint.com

Object Oriented Programming using Java Laboratory (DJS23FLES201)
Academic Year 2023-24