OOP WITH JAVA
LAB MANUAL
COURSE CODE: 15CS46P

FOR 4th Sem CS & E
(2017-18)

BY
Mr. SUBHASH J R
LECTURER
COMPUTER SCIENCE & ENGINEERING
RJS POLYTECHNIC
BANGALORE-34.

1. WRITE A JAVA PROGRAM TO SORT LIST OF NAMES USING SELECTION SORT TECHNIQUE

```
class pgm1
{
        public static void main(String JINKA[])
        {
                String names[]={"ranganath","lakshmidevi","subhash","reshma",
                                "suma","prasanna","saiprem","sushruth"};
                for(int i=0;i<names.length;i++)
                {
                        int index=i;
                        for(int j=i+1;j<names.length;j++)
                        if(names[j].compareTo(names[index])<0)
                        index=j;
                        String temp=names[index];
                        names[index]=names[i];
                        names[i]=temp;
                }
                System.out.println("Names in the sorted listed are displayed below");
                for(int i=0;i<names.length;i++)
                System.out.println("    " +names[i]);
        }
}
```

OUTPUT:

```
E:\subhash>javac pgm1.java
E:\subhash>java pgm1
Names in the sorted listed are displayed below
    lakshmidevi
    prasanna
    ranganath
    reshma
    saiprem
    subhash
    suma
    sushruth
```

2. WRITE A JAVA PROGRAM TO DEFINE A CLASS, CONSTRUCTOR AND OVERLOAD THE CONSTRUCTOR.

```
class student
{
        String name,location;
        int postid;
        student()
        {
                name="Spoorthi P M";
                location="dvg";
                postid=577003;
        }
        student(String name,String location,int postid)
        {
                this.name=name;
                this.location=location;
                this.postid=postid;
        }
        void display()
        {
                System.out.println(name   +"\t"+location  +"\t" +postid);
        }
}
class pgm2
{
        public static void main(String  b[])
        {
                student  s1= new student();
                student s2=new student("pammu","davanagere", 577005);
                s1.display();
                s2.display();
        }
}
```

OUTPUT:

```
E:\subhash>javac pgm2.java
E:\subhash>java pgm2
Spoorthi P M    dvg     577003
pammu   davanagere      577005
```

3. WRITE A JAVA PROGRAM TO DEFINE A CLASS, DEFINE INSTANCE METHODS FOR SETTING AND RETRIEVING VALUES OF INSTANCE VARIABLES.

```
class student
{
        String name,reg_no,branch,college;
        int sem;
        void setdata(String n,String r,String b,String c,int s)
        {
                name=n;
                reg_no=r;
                branch=b;
                college=c;
                sem=s;
        }
        void retrieve_data()
        {
                System.out.println("My name is:" +name);
                System.out.println("My register number is:" +reg_no);
                System.out.println("I am studying in the branch of :" +branch);
                System.out.println("My college name is:" +college);
                System.out.println("I am studying in semester:" +sem);
        }
}

class pgm3
{
        public static void main(String c[])
        {
                student s=new student();
                s.setdata("crazy","372cs18000","computer science", "RJS POLYTECHNIC Koramangala
                                                Bengaluru", 1);
                s.retrieve_data();
        }
}
```

OUTPUT:

My name is:crazy
My register number is:372cs18000
I am studying in the branch of :computer science
My college name is:RJS POLYTECHNIC Koramangala Bengaluru
I am studying in semester:1

4. WRITE A JAVA PROGRAM TO DEFINE A CLASS, DEFINE INSTANCE METHODS AND OVERLOAD THEM AND USE THEM FOR DYNAMIC METHOD INVOCATION.

```
class sum
{
	void display(int a,int b)
	{
		int result=a+b;
		System.out.println("the sum of 2 integers are:" +result);
	}
	void display(double a,double b)
	{
		double result=a+b;
		System.out.println("the sum of 2 floating numbers are:" +result);
	}
	void display(int a,int b,int c)
	{
		int result=a+b+c;
		System.out.println("the sum of 3 integers are:" +result);
	}
	void display(int a,double b)
	{
		double result=a+b;
		System.out.println("the sum an integer with a floating number is:" +result);
	}
}
class pgm4
{
	public static void main(String d[])
	{
		sum obj=new sum();
		obj.display(5,8);
		obj.display(5.11,8.43);
		obj.display(5,8,4);
		obj.display(5,8.0132);
	}
}
```

OUTPUT:

the sum of 2 integers are:13
the sum of 2 floating numbers are:13.54
the sum of 3 integers are:17
the sum an integer with a floating number is:13.0132

5. WRITE A JAVA PROGRAM TO DEMONSTRATE USE OF SUB CLASS.

```
class  rjsp
{
        String branch;
        void get_branch(String b)
        {
                branch=b;
        }
        void show_branch()
        {
        System.out.println("I am studying in  RJS POLYTECHNIC, " + branch + " Branch");
        }
}
class comscience extends rjsp
{
        String location;
        void get_loc(String l)
        {
                location=l;
        }
        void show_loc()
        {
                System.out.println("Located in " + location +" Bengaluru");
        }
}
class pgm5
{
        public static void main(String e[])
        {
                comscience cs = new comscience();
                cs.get_branch("COMPUTER SCIENCE & ENGG.");
                cs.show_branch();
                cs.get_loc("KORAMANGALA");
                cs.show_loc();
        }
}
```

OUTPUT:

```
E:\subhash>javac pgm5.java
E:\subhash>java pgm5
I am studying in  RJS POLYTECHNIC, COMPUTER SCIENCE & ENGG. Branch
Located in KORAMANGALA Bengaluru
```

6. WRITE A JAVA PROGRAM TO DEMONSTRATE NESTED CLASS.

```
class krjs
{
        String name="KARNATAKA REDDY JANA SANGHA";
        class rjs
        {
                String name1=" RJS POLYTECHNIC";
                void show()
                {
                        System.out.println(name +"\t" +name1);
                }
        }
}
class pgm6
{
        public static void main(String f[])
        {
                krjs obj1=new krjs();
                krjs.rjs obj2=obj1.new rjs();
                obj2.show();
        }
}
```

OUTPUT:

```
E:\subhash>javac pgm6.java
E:\subhash>java pgm6
KARNATAKA REDDY JANA SANGHA RJS POLYTECHNIC
```

7. WRITE A JAVA PROGRAM TO IMPLEMENT ARRAY OF OBJECTS.

```
class employee
{
	String name,designation;
	employee(String n,String d)
	{
		name=n;
		designation=d;
	}
	void show_data()
	{
		System.out.println(name +"\t" +designation);
	}
}

class pgm7
{
	public static void main(String g[])
	{
		employee[] staff= new employee[4];
		staff[0]=new employee("vishal","jun engg.");
		staff[1]=new employee("karthik","jun engg.");
		staff[2]=new employee("radha","asst engg.");
		staff[3]=new employee("murugesh","exe chief engg.");
		for(int i=0;i<=3;i++)
		staff[i].show_data();
	}
}
```

OUTPUT:

```
E:\subhash>javac pgm7.java
E:\subhash>java pgm7
vishal  jun engg.
karthik jun engg.
radha   asst engg.
murugesh        exe chief engg.
```

8A. WRITE A JAVA PROGRAM TO PRACTICE METHODS OF STRING CLASS.

```
class pgm8a
{
	public static void main(String i[])
	{
		String s1="black duster";
		String s2="green board";
		String s3="chalk piece";
		System.out.println("length of string s1 is: "  + s1.length());
		System.out.println("the character present in the position3 of string s2 is:"  +
                                                                    s2.charAt(3));
		System.out.println("the number of occurance of k in s3 is:"  + s3.indexOf('k'));
		String s4="is not a car";
		String s5="xuv";
		String s6="XUV";
		System.out.println("concatination of 2 strings done here:" + s1.concat(s4));
		System.out.println("equals method demonstartion shown here:"  + s5.equals(s6));
		System.out.println("equals ignore case  method demonstartion shown here:"  +
                                                        s5.equalsIgnoreCase(s6));
		System.out.println("case demonstration shown here:" + s1.toUpperCase());
		System.out.println("case demonstration shown here:" + s1.toLowerCase());
		String s7= "       see the difference";
		System.out.println("before trim method string s7 looks like this: " + s7);
		System.out.println("before trim method string s7 looks like this: " + s7.trim());
	}
}
```

OUTPUT:

```
E:\subhash>javac pgm8a.java
E:\subhash>java pgm8a
length of string s1 is: 12
the character present in the position3 of string s2 is:e
the number of occurance of k in s3 is:4
concatination of 2 strings done here:black dusteris not a car
equals method demonstartion shown here:false
equals ignore case  method demonstartion shown here:true
case demonstration shown here:BLACK DUSTER
case demonstration shown here:black duster
before trim method string s7 looks like this:       see the difference
before trim method string s7 looks like this: see the difference
```

8B. WRITE A JAVA PROGRAM TO DEMONSTRATE METHODS OF STRING BUFFER CLASS.

```
class pgm8b
{
        public static void main(String j[])
        {
                StringBuffer sb=new StringBuffer("this is my collage ");
                System.out.println("the string sb contains : " +sb);
                System.out.println("length of the string sb contains :" +sb.length()  +" including white
                                                                                spaces ");
                System.out.println("capacity of the string is :" +sb.capacity());
                System.out.println("the character at an index of 6 is :" +sb.charAt(6));
                sb.setCharAt(15,'e');
                System.out.println("after setting character at 15th position by e is:" +sb);
                System.out.println("after appending string with existing string buffer is:" +sb.append("
                                                                        in bengaluru"));
                System.out.println("after inserting  method:" +sb.insert(18," RJS"));
                System.out.println("after delete method:" +sb.delete(11,18));
        }
}
```

OUTPUT:

```
E:\subhash>javac pgm8b.java
E:\subhash>java pgm8b
the string sb contains : this is my collage
length of the string sb contains :19 including white spaces
capacity of the string is :35
the character at an index of 6 is :s
after setting character at 15th position by e is:this is my college
after appending string with existing string buffer is:this is my college  in ben
galuru
after inserting  method:this is my college RJS  in bengaluru
after delete method:this is my  RJS  in bengaluru
```

9. WRITE A JAVA PROGRAM TO IMPLEMENT VECTOR CLASS AND ITS METHODS.

```
import java.util.*;
class pgm9
{
	public static void main(String k[])
	{
		Vector v=new Vector();
		v.addElement("bcs");
		v.addElement("c");
		v.addElement("ds");
		v.insertElementAt("oop with java",0);
		v.insertElementAt("c++",3);
		v.insertElementAt("pch/w & netw",5);
		v.insertElementAt("linux",5);
		System.out.println("vector size is:" +v.size());
		System.out.println("capacity of vector is:" +v.capacity());
		System.out.println("the first element in the vector is:" +v.firstElement());
		System.out.println("the last element in the vector is:" +v.lastElement());
		System.out.println("the element oop with java found at the position :" +v.indexOf("oop
                                                                        with java"));
		v.removeElement("ds");
		v.removeElementAt(4);
		System.out.println("after removing 2 elements size is:" + v.size());
		System.out.println("the elements of the vector are\n");
		for(int i=0;i<v.size();i++)
		System.out.println(v.elementAt(i)+ " ");
	}
}
```

OUTPUT:

```
E:\subhash>javac pgm9.java
Note: pgm9.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
E:\subhash>java pgm9
vector size is:7
capacity of vector is:10
the first element in the vector is:oop with java
the last element in the vector is:pch/w & netw
the element oop with java found at the position :0
after removing 2 elements size is:5
the elements of the vector are
oop with java
bcs
c
c++
pch/w & netw
```

10. WRITE A JAVA PROGRAM TO IMPLEMENT WRAPPER CLASS AND THEIR METHODS.

```
class pgm10
{
	public static void main(String l[])
	{
		System.out.println("creating an object for value 10");
		Integer i= new Integer(83);
		System.out.println("obtaining the value back from the object:" +i.intValue());
		System.out.println("creating an object for value 10.4");
		Float f= new Float(10.4);
		System.out.println("obtaining the value back from the object:" +f.floatValue());
		System.out.println("creating an object for value S");
		Character c= new Character('S');
		System.out.println("obtaining the value back from the object:" +c.toString());
	}
}
```

OUTPUT:

```
E:\subhash>javac pgm10.java
E:\subhash>java pgm10
creating an object for value 10
obtaining the value back from the object:83
creating an object for value 10.4
obtaining the value back from the object:10.4
creating an object for value S
obtaining the value back from the object:S
```

11. WRITE A JAVA PROGRAM TO IMPLEMENT INHERITANCE AND DEMONSTRATE USE OF METHOD OVERRIDING.

```
class A
{
        void hello()
        {
                System.out.println("hello i am in parent class");
        }
}

class B
{
        void hello()
        {
                System.out.println("hello i am overridden");
        }
}
class pgm11
{
        public static void main(String m[])
        {
                B b=new B();
                b.hello();
        }
}
```

OUTPUT:

```
E:\subhash>javac pgm11.java
E:\subhash>java pgm11
hello i am overridden
```

12. WRITE A JAVA PROGRAM TO IMPLEMENT MULTILEVEL INHERITANCE BY APPLYING VARIOUS ACCESS CONTROLS TO ITS DATA MEMBERS AND METHODS.

```
class A
{
	int a;
	void set_a(int a)
	{
		this.a=a;
	}
	void show_a()
	{
		System.out.println("value of a is :"+a);
	}
}
class B extends A
{
	int b;
	void set_b(int b)
	{
		this.b=b;
	}
	void show_b()
	{
		System.out.println("value of b is :"+b);
	}
}
class C extends B
{
	int c;
	void set_c(int c)
	{
		this.c=c;
	}
	void show_c()
	{
		System.out.println("value of c is :"+c);
	}
}
```

```
class pgm12
{
        public static void main(String n[])
        {
                C c=new C();
                c.set_a(10);
                c.set_b(4);
                c.set_c(83);
                c.show_a();
                c.show_b();
                c.show_c();
        }
}
```

OUTPUT:

```
E:\subhash>javac pgm12.java
E:\subhash>java pgm12
value of a is :10
value of b is :4
value of c is :83
```

13A. WRITE A JAVA PROGRAM TO DEMONSTRATE USE OF IMPLEMENTING INTERFACES.

```
interface area
{
        final static float pi=3.14F;
        float compute(float x,float y);
}

class rectangle implements area
{
        public float compute(float x,float y)
        {
                return(x*y);
        }
}
class circle implements area
{
        public float compute(float x,float y)
        {
                return(pi*x*x);
        }
}

class pgm13a
{
        public static void main(String o[])
        {
                rectangle rect=new rectangle();
                circle cir =new circle();
                area a;
                a=rect;
                System.out.println("the area of rectangle is:"+a.compute(5,10));
                a=cir;
                System.out.println("the area of rectangle is:"+a.compute(10,10));
        }
}
```

OUTPUT:

```
E:\subhash>javac pgm13a.java
E:\subhash>java pgm13a
the area of rectangle is:50.0
the area of rectangle is:314.0
```

13B. WRITE A JAVA PROGRAM TO DEMONSTARTE EXTENDING INTERFACES.

```
interface area
{
        final static float pi=3.14F;
        double compute(double x,double y);
}
interface display extends area
{
        void display_result(double result);
}
class rectangle implements display
{
        public double compute(double x,double y)
        {
                return(x*y);
        }
        public void display_result(double result)
        {
                System.out.println("the area is :"+result);
        }
}
class pgm13b
{
        public static void main(String p[])
        {
                rectangle rect=new rectangle();
                double result=rect.compute(10.4,19.83);
                rect.display_result(result);
        }
}
```

OUTPUT:

```
E:\subhash>javac pgm13b.java
E:\subhash>java pgm13b
the area is :206.232
```

14. WRITE A JAVA PROGRAM TO IMPLEMENT THE CONCEPT OF IMPORTING CLASSES FROM USER DEFINED PACKAGE, CREATING PACKAGES.

Create a folder mypack---->myself.java
In this following code should be written and saved

```
package mypack;
public class myself
{
	String name,dept;
	int sem;
	public void get_info(String n,String d,int s)
	{
		name=n;
		dept=d;
		sem=s;
	}
	public void show_info()
	{
		System.out.println("my name is :" +name);
		System.out.println("I am studying in  :" +dept);
		System.out.println("now i am in semester :" +sem);
	}
}
```

Below program accessing the above package program stored in different location

```
import mypack.myself;
class pgm14
{
	public static void main(String q[])
	{
		myself obj1=new myself();
		myself obj2=new myself();
		obj1.get_info("Varun P Y","commerce",1);
		obj2.get_info("Karthik PM","science",1);
		obj1.show_info();
		obj2.show_info();

	}
}
```

OUTPUT:

```
E:\subhash>cd mypack
E:\subhash\mypack>set path=E:\CSE Softwares\sel\jdk1.6.0_25\bin
E:\subhash\mypack>javac myself.java
E:\subhash\mypack>cd..
E:\subhash>javac pgm14.java
E:\subhash>java pgm14
my name is :Varun P Y
I am studying in  :commerce
now i am in semester :1
my name is :Karthik PM
I am studying in  :science
now i am in semester :1
```

15A. WRITE A JAVA PROGRAM TO IMPLEMENT THE CONCEPT OF THREADING BY EXTENDING THREAD CLASS

```
class one extends Thread
{
	public void run()
	{
		System.out.println("thread one starts here");
		for(int i=1;i<=5;i++)
		{
			System.out.println("from thread one: i= "+i);
			if(i==1)
			yield();
		}
		System.out.println("exiting from one");
	}
}
class two extends Thread
{
	public void run()
	{
		System.out.println("thread two starts here");
		for(int j=1;j<=5;j++)
		{
			System.out.println("from thread two: j= "+j);
			if(j==3)
			try
			{
				sleep(3000);
			}
			catch(Exception e)
			{
				System.out.println("sleep for 3ms");
			}
		}
		System.out.println("exiting from two");
	}
}
class three extends Thread
{
```

```
	public void run()
	{
		System.out.println("thread three starts here");
		for(int k=1;k<=5;k++)
		{
			System.out.println("from thread three: k= " +k);
			if(k==2)
			stop();
		}

		System.out.println("exiting from three");
	}
}

class pgm15a
{
	public static void main(String r[])
	{
		one t1=new one();
		two t2=new two();
		three t3=new three();
		t1.start();
		t2.start();
		t3.start();
	}
}
```

OUTPUT:

```
E:\subhash>javac pgm15a.java
Note: pgm15a.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\subhash>java pgm15a
thread one starts here
thread three starts here
thread two starts here
from thread three: k= 1
from thread one: i= 1
from thread three: k= 2
from thread two: j= 1
from thread one: i= 2
from thread two: j= 2
from thread one: i= 3
from thread two: j= 3
from thread one: i= 4
from thread one: i= 5
exiting from one
from thread two: j= 4
from thread two: j= 5
exiting from two
```

15B. WRITE A JAVA PROGRAM TO DEMONSTRATE THREAD USING RUNNABLE INTERFACE

```
import java.lang.*;
class mythread implements Runnable
{
	public void run()
	{
		System.out.println("creating thread using runnable interface");
		for(int i=1;i<=10;i++)
		{
			System.out.println("PADMASHALI - JINKA" +i);
		}
		System.out.println("End of thread");
	}
}
class pgm15b
{
	public static void main(String s[])
	{
		mythread x=new mythread();
		Thread th=new Thread(x);
		th.start();
	}
}
```

OUTPUT:

```
E:\subhash>set path=E:\CSE Softwares\sel\jdk1.6.0_25\bin
E:\subhash>javac pgm15b.java
E:\subhash>java pgm15b
creating thread using runnable interface
PADMASHALI - JINKA1
PADMASHALI - JINKA2
PADMASHALI - JINKA3
PADMASHALI - JINKA4
PADMASHALI - JINKA5
PADMASHALI - JINKA6
PADMASHALI - JINKA7
PADMASHALI - JINKA8
PADMASHALI - JINKA9
PADMASHALI - JINKA10
End of thread
```

16A. WRITE A JAVA PROGRAM TO IMPLEMENT THE CONCEPT OF EXCEPTION HANDLING USING PREDEFINED EXCEPTION.

```
class pgm16a extends Exception
{
	public static void main(String t[])
	{
		int a=10,b=0,c;
		try
		{
			c=a/b;
		}
		catch(ArithmeticException e)
		{
			System.out.println(e.getMessage());
		}
	}
}
```

OUTPUT:

```
E:\subhash>javac pgm16a.java
E:\subhash>java pgm16a
/ by zero
```

16B. WRITE A JAVA PROGRAM TO ILLUSTRATE THROWING OUT OF USER DEFINED EXCEPTION

```
class myownexception extends Exception
{
	myownexception(String msg)
	{
		super(msg);
	}
}
class pgm16b
{
	public static void main(String u[])
	{
		int age=15;
		try
		{
			if(age<18)
			throw new myownexception("your age is less than the actual age to vote");
		}
		catch(myownexception e)
		{
			System.out.println("this is my exception block");
			System.out.println(e.getMessage());
		}
		finally
		{
			System.out.println("finally block: End of program");
		}
	}
}
```

OUTPUT:

```
E:\subhash>java pgm16b
this is my exception block
your age is less than the actual age to vote
finally block: End of program
```