



Week 1

Write a program by creating an 'Employee' class having the following methods and print the final salary.

1 - 'getInfo()' which takes the input as name, id, salary, number of hours of work per day of employee.

2 - 'AddSal()' which adds \$10 to salary of the employee if it is less than \$500

3 - 'AddWork()' which adds \$5 to salary of employee if the number of hours of work per day is more than 6 hours.

```
import java.util.*;
```

```
class Employee {
```

```
    String name;
```

```
    int id;
```

```
    double salary, work_hr;
```

```
    void getInfo() {
```

```
        Scanner read = new Scanner(System.in);
```

```
        System.out.print("Enter name: ");
```

```
        name = read.next();
```

```
        System.out.print("Enter ID: ");
```

```
        id = read.nextInt();
```

```
        System.out.print("Enter Salary: ");
```

```
        salary = read.nextDouble();
```

```
        System.out.print("No. of Work hours: ");
```

```
        work_hr = read.nextDouble();
```

```
}
```

```
    void addSal() {
```

```
        if (salary < 500) salary += 10; }
```

```

void addwork() {
    if (work-hr > 6)
        salary += 5;
}

void details() {
    System.out.println("Name: " + name);
    System.out.println("Id: " + id);
    System.out.println("Salary : " + salary);
    System.out.println("No.of hours : " + work-hr);
}

public class Emp {
    public static void main(String args[]) {
        Scanner read = new Scanner(System.in);
        System.out.print("No.of Employees: ");
        int n = read.nextInt();
        Employee obj[] = new Employee[n];
        for (int i = 0; i < n; i++) {
            obj[i] = new Employee();
            obj[i].getinfo();
            obj[i].addsal();
            obj[i].addwork();
        }
        for (int j = 0; j < n; j++)
            obj[j].details();
    }
}

```

Week 1

i) Write a java program to demonstrate a static keyword.

```
import java.util.*;
public class Taking{
    static String name, gender;
    static int id, age;
    //char gender;
    static void reading(){
        Scanner read = new Scanner(System.in);
        System.out.print("Name : ");
        name = read.next();
        System.out.print("Id : ");
        id = read.nextInt();
        System.out.print("Age = ");
        age = read.nextInt();
        System.out.print("gender = ");
        gender = read.next();
    }
    static void details(){
        System.out.println("Name : " + name);
        System.out.println("Id : " + id);
        System.out.println("Age : " + age);
        System.out.println("Gender : " + gender);
    }
    public static void main(String args[]){
        reading();
        details();
    }
}
```

iii) Write a java program to demonstrate this method

class Demo{

    double l,b,h;

    Demo(double l, double b, double h){

        this.l=l;

        this.b=b;

        this.h=h;

    double area(){

        return 2\*(l\*b+b\*h+h\*l);

}

    double volume(){

        return l\*b\*h;

}

class Onconstruct{

    public static void main(String[] args){

        Demo obj=new Demo(10,20,30);

        double arr;

        arr=obj.area();

        double vall=obj.volume();

        System.out.println("Area of cuboid = "+arr);

        System.out.println("Volume of cuboid = "+vall);

}

2

iii) Write a java program to demonstrate a variable length arguments.

```

import java.util.*;
class Varargpro{
    static void varstr(int ...a){
        System.out.println("No. of arguments : "+a.length);
        for (int i : a)
            System.out.print(i+" ");
        System.out.println();
    }
    static void varstr (String ...a){
        System.out.println("No. of arguments : "+a.length);
        for (String i : a)
            System.out.print(i+" ");
        System.out.println();
    }
    static void varstr (String name){
        System.out.println("Welcome "+name);
    }
    public static void main(String args[]){
        varstr(1,2,3,4);
        varstr();
        varstr("manoj", "ravi", "manu");
        System.out.println("Name : ");
        Scanner read = new Scanner(System.in);
        String name = read.nextLine();
        varstr(name);
    }
}

```

Ques

Week 3

Write a program for the following: an inner class named Inner is defined within the scope of class Outer. Therefore, any code in class Inner can directly access the variable Outer\_x. An instance method named display() is defined inside Inner. This method displays outer\_x on the standard output stream. The main() method of InnerClassDemo creates an instance of class Outer and invokes its test() method. That method creates an instance of class Inner and the display() method is called.

```

import java.util.*;
class Outer{
    Outer(int name){
        System.out.println("[In Constructor]");
        if(name==2)
            Inner obj2 = new Inner();
            obj2.display();
        else
            test();
    }
    static void test(){
        System.out.println("[In Outer Class]");
        System.out.println("All about student details ...");
        System.out.println("_____\n_____\n");
    }
    class Inner{
        void display(){
    }
}

```

```
System.out.println (" [In Inner Class] ");
System.out.println (" All about Staff details ... ");
System.out.println (" ---In---In---In ");
test();
}
```

2  
class Innerclass {

```
public static void main (String[] args) {
    System.out.print (" Student(1) or Staff(2) : ");
    Scanner read = new Scanner (System.in);
    int name = read.nextInt ();
    Outer obj = new Outer (name);
}
```

3

ii) Write a java program that displays the time in different formats in the form of HH, MM, SS using constructor overloading.

```

import java.util.*;
class Overload {
    int hh, mm, ss;
    Overloads() {
        hh = 5; mm = 23; ss = 57;
        System.out.println("[Constructor Overloading, no parameter]");
        System.out.println(hh + ":" + mm + ":" + ss);
    }
    Overload(int hh) {
        mm = 28; ss = 55;
        System.out.println("[Constructor overloading, 1 parameter]");
        System.out.println(hh + ":" + mm + ":" + ss);
    }
    Overload(int hh, int mm) {
        ss = 55;
        System.out.println("[Constructor overloading, 2 parameters]");
        System.out.println(hh + ":" + mm + ":" + ss);
    }
    Overload(int hh, int mm, int ss) {
        System.out.println("[Constructor overloading, 3 parameters]");
        System.out.println(hh + ":" + mm + ":" + ss);
    }
}

```

```
class Overload_time{
```

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

```
        Scanner read = new Scanner(System.in);
```

```
        System.out.println("No. of arguments: ");
```

```
        int num = read.nextInt();
```

```
        int hh, mm, ss;
```

```
        switch (num) {
```

Case 0:

```
        Overload obj = new Overload();
```

```
        break;
```

Case 1:

```
        System.out.print("Hours : ");
```

```
        hh = read.nextInt();
```

```
        Overload obj2 = new Overload(hh);
```

```
        break;
```

Case 2:

```
        System.out.print("Hrs, min = ");
```

```
        hh = read.nextInt();
```

```
        mm = read.nextInt();
```

```
        Overload obj2 = new Overload(hh, mm);
```

```
        break;
```

Case 3:

```
        System.out.print("Hrs, min, sec = ");
```

```
        hh = read.nextInt();
```

```
        mm = read.nextInt();
```

```
        ss = read.nextInt();
```

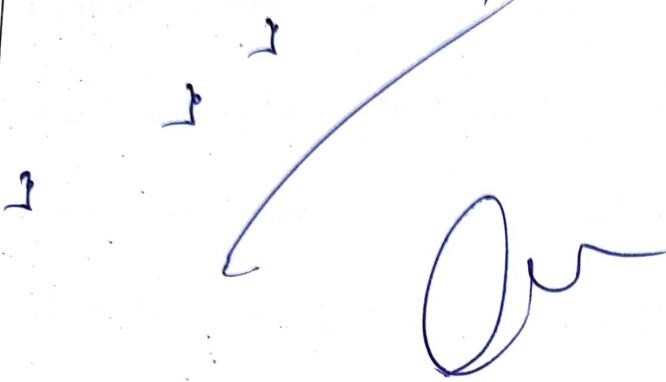
```
        Overload obj3 = new Overload(hh, mm, ss);
```

```
        break;
```

default:

System.out.println ("Wrong Input");

break;



Week 4

Write a Java program for the following: create class employee which takes input as name, id, designation & create another class salary which takes input as BP, HRA, DA, PF and inherits the members of class employee. Print Net pay of each employee using reademp() in employee class, readsalary(), calculateSalary() & displayemp() methods in salary class.

```

import java.util.*;
class Employee{
    String name,desig;
    long id;
    void reademp(){
        System.out.println("In Employee (Base class)");
        Scanner readn=new Scanner(System.in);
        System.out.print("Name: ");
        name=readn.next();
        System.out.print("Id: ");
        id=readn.nextLong();
        System.out.print("Designation: ");
        desig=readn.next();
    }
    void printit(){
        System.out.println("Name: "+name);
        System.out.println("Id: "+id);
    }
}

```

System.out.println ("Designation: "+desig);

class Salary extends Employee {

double BP, HRA, DA, PF, NP;

void readsalary() {

System.out.println ("In Salary (Derived class)");

Scanner read = new Scanner (System.in);

System.out.print ("Enter Basic pay: ");

BP = read.nextDouble();

System.out.print ("HRA: ");

HRA = read.nextDouble();

System.out.print ("DA: ");

~~DA = read.nextDouble();~~

System.out.print ("PF: ");

~~PF = read.nextDouble();~~

void calculatesalary() {

NP = BP + HRA + DA - PF;

void displayemp() {

System.out.println (NP);

```
class Emp inherit {  
    public static void main(String args[]) {  
        System.out.println("Main class");  
        Scanner readit = new Scanner(System.in);  
        System.out.print("No. of Employees : ");  
        int n = readit.nextInt();  
        Salary obj[] = new Salary[n];  
        for (int i = 0; i < n; i++) {  
            System.out.println("Enter details");  
            obj[i] = new Salary();  
            obj[i].reademp();  
            obj[i].readsalary();  
            obj[i].calculatesalary();  
        }  
    }  
}
```

```
for (int i = 0; i < n; i++) {  
    System.out.print(obj[i].name);  
    System.out.print(" salary ");  
    obj[i].displayemp();  
}
```

3

3

Om

ii. Write a java program for the following. Assume that the test results of a batch of students are stored in three different classes. Class student stores the name, roll number and class test stores the marks obtained in six subjects & class result contains the total marks obtained in test. The class result can inherit the details of marks obtained in test & name, roll number of students through multi level inheritance

```

import java.util.*;
class Student {
    String name;
    int rno;
}

class Test extends Student {
    int arr[] = new int[6];
}

class Result extends Test {
    static int s=0;
    void reading() {
        Scanner readn = new Scanner(System.in);
        System.out.print("Enter name: ");
        name = readn.next();
        System.out.print("R.no: ");
        rno = readn.nextInt();
    }
}

```

```

System.out.print ("Marks : ");
for (int j=0; j<6; j++)
    arr[j] = read.nextInt();
}

void sumi()
{
    for (int j=0; j<6; j++)
        s += arr[j];
}

void display()
{
    System.out.println ("Name : " + name + " In R.no : " + rno);
    System.out.println ("Total : " + s);
}

class Student_Marks
{
    public static void main (String args[])
    {
        Scanner read = new Scanner (System.in);
        System.out.print ("No. of Students : ");
        int n = read.nextInt();
        result obj[] = new result[n];
        for (int i=0; i<n; i++)
        {
            obj[i] = new result();
            obj[i].reading();
            obj[i].sumi();
        }
        for (int j=0; j<n; j++)
            obj[j].display();
    }
}

```

iii, Write a Java program for the following: employee is the parent class which is common for all sub or child classes both permanent-employee class & temporary-employee class. Use read-emp() method in employee class which reads name, id, salary. And use print-emp() method in both classes which print details.

```
import java.util.*;
class Employee{
    String name;
    int id;
    double salary;
    void read_emp(){
        Scanner readn=new Scanner(System.in);
        System.out.print("Name : ");
        name=readn.next();
        System.out.print("Enter Id : ");
        id=readn.nextInt();
        System.out.print("Enter Salary : ");
        salary=readn.nextDouble();
    }
}
```

3  
class Permanent-class extends Employee{

```
void print_emp(){
    read_emp();
    System.out.println("Name: "+name);
```

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

```
double val = (s/100)*salary;
```

```
salary = salary + val;
```

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

```
class Temper-class extends Employee {
```

```
void printEmp() {
```

```
readEmp();
```

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

```
double val = (3.5/100)*salary;
```

```
salary += val;
```

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

```
public class Salary {
```

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

```
permanent-class obj1 = new Permanent-class();
```

```
Temper-class obj2 = new Temper-class();
```

```
int n = read.nextInt();
```

```
for (int i=0; i<n; i++) {
```

```
System.out.print("Select : ");
```

```
System.out.print("Permanent (1) or Temp (2)");
```

```
int opt = read.nextInt();
```

switch(opt){

case 1:

    obj1.print\_emp();

    break;

case 2:

    obj2.print\_emp();

    break;

default:

    System.out.println("Wrong Input");

}

3

✓ Dr

Week 5:

↓ Write a Java program to compute area of room, using method overloading. (Area of room = length × Breadth) (Consider method with no parameters, method with single parameter & method with two parameters.)

class program{

```
    int length;
    int breadth;
    void reading(){
```

length=10;

breadth=20;

}

void reading(int l){

length=l;

breadth=20;

}

void reading(int l, int b){

length=l;

breadth=b;

}

int area(){

return length \* breadth;

}

3  
class Overloading\_pro{

public static void main(String[] args) {

int x;

program obj=new program();

System.out.println("method with no parameters");

```
obj.reading();
x=obj.area();
System.out.println(x);
System.out.println("method with one parameter");
obj.reading(5);
x=obj.area();
System.out.println(x);
System.out.println(" method with two parameters");
obj.reading(5,5);
x=obj.area();
System.out.println(x);
```

{

ii) Write a program to create super class called Figure that stores the dimensions of two-dimensional object. It also defines a method called area() that computes the area of an object. The program derives two subclasses from fig. The first is rectangle and second is Triangle. Each of these subclasses overrides area() so that it returns the area of rectangle & triangle.

class figure{

```
int a=5;
int b=15;
double area;
```

void area(){

```
area=(double)a*b;
System.out.println(area);
```

}

class rectangle extends figure{

```
int a=10, b=20;
```

void area(){

```
area=(double)a*b;
```

```
System.out.println(area);
```

}

class triangle extends figure{

```
int a=5, b=15;
```

void area(){

```
area=(double)a*b;
```

```
System.out.println(area); }
```

class lab-5b{

    public static void main (String args[]){

        rectangle obj=new rectangle();

        System.out.print ("Area of rectangle : ");

        obj.area();

        triangle obj2=new triangle();

        System.out.print ("Area of triangle : ");

        obj2.area();

3

✓ BB

iii) Write a Java program to demonstrate a final keyword.

```

class Demo {
    final int a = 20;
    void display() {
        System.out.print(" Value of fixed a : " + a);
        // a += 5;
        // System.out.print (" Throws error : " + a);
    }
}

class Java_sc {
    public static void main (String [] args) {
        Demo obj = new Demo();
        obj.display();
    }
}

```

(B)

Week 6.

i) Write a Java program to create an abstract class named shape that contains two integers and an empty method named printArea(); Provide 3 classes named Rectangle, Triangle, Circle such that each one of the classes extends class shape. Each one of classes contains only method printArea() that prints area of given shape.

abstract class Shape{

    double a,b;

    abstract void printName();

}

class Rectangle extends Shape{

    void printArea(){

        a=20;

        System.out.print("Area of Rectangle "+(a\*b));

}

class Triangle extends Shape{

    void printArea(){

        a=10

        b=20,

        System.out.println("Area of Triangle : "+(a\*b\*0.5));

}

3

class Circle extends Shape {

void paintArea() {

$$\alpha = 5,$$

System.out.println("Area of Circle : " + (3.14 \* alpha \* alpha));

{}

class java\_lab6\_1 {

public static void main(String[] args) {

Rectangle obj = new Rectangle();  
obj.paintArea();

Triangle obj2 = new Triangle();  
obj2.paintArea();

Circle obj3 = new Circle();

obj3.paintArea();

{}

B2

Write a java program to find details of students eligible to enroll for exam using interfaces

```
import java.util.*;
```

```
interface Student {
```

```
    abstract String readval();
```

```
interface Department {
```

```
    abstract void attend();
```

```
class Exam implements Student, Department
```

```
Scanner read = new Scanner(System.in);
```

```
int c, aten, no, clss;
```

```
String name;
```

```
public String readval() {
```

```
    System.out.println("Roll no:");
```

```
    no = read.nextInt();
```

```
    System.out.println("Class no:");
```

```
    clss = read.nextInt();
```

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

```
    name = read.next();
```

```
    return name;
```

```
public void attend()
```

```
    System.out.println("Enter attendance(%): ");
```

```
    aten = read.nextInt();
```

```
public void calculate() {
    if (atm > 75)
        c++;
}

public Boolean eligible() {
    if (c == 1)
        return true;
    return false;
}

public class lab6_2 {
    public static void main(String[] args) {
        Exam obj = new Exam();
        String name = obj.readval();
        obj.attend();
        obj.calculate();
        Boolean get = obj.eligible();
        if (get)
            System.out.println("name + " is Eligible");
        else
            System.out.println("name + " is not eligible");
    }
}
```

## Week 7: String Methods

```

import java.util.*;
public class problems_str {
    public static void main(String[] args) {
        Scanner read = new Scanner(System.in);
        System.out.print("Enter string 1:");
        String a = read.nextLine();
        System.out.print("Enter string 2:");
        String b = read.nextLine();
        System.out.println("1. length\n2. Compare\n3. Concatenate");
        System.out.println("4. To lower\n5. To upper\n6. exit");
        System.out.println("Select an option:");
        int opt = read.nextInt();
        while (opt != 6) {
            switch (opt) {
                case 1:
                    System.out.println("length of a: " + a.length());
                    System.out.println("length of b: " + b.length());
                    break;
                case 2:
                    if (a.compareTo(b) < 0)
                        System.out.println(a + " < " + b);
                    else if (a.compareTo(b) > 0)
                        System.out.println(a + " > " + b);
                    else
                        System.out.println(a + " = " + b);
                    break;
                case 3:
                    System.out.println("Concatenated strings: " + (a.concat(b)));
                    break;
            }
        }
    }
}

```

Case 4:

```
System.out.println("Str a is lower: "+a.toLowerCase(casec));
```

```
System.out.println("Str b is lower: "+b.toLowerCase(casec));  
break;
```

Case 5:

```
System.out.println("Str d is Upper: "+d.toUpperCase(casec));
```

```
System.out.println("Str b is Upper: "+b.toUpperCase(casec));  
break;
```

Case 6:

```
break;
```

default:

```
System.out.println("Wrong Input");
```

3

```
System.out.println("Select an option: ");
```

```
opt = read.nextInt();
```

↙  
④

4

}

```

import java.util.*;
public class program_on_strbuf {
    public static void main(String[] args) {
        Scanner read = new Scanner(System.in);
        System.out.println("String a:");
        String a = read.nextLine();
        System.out.println("String b:");
        String b = read.nextLine();
        StringBuffer aa = new StringBuffer(a);
        StringBuffer bb = new StringBuffer(b);
        System.out.println("1.length\n2.reverse\n3.append\n4.findIndex");
        System.out.println("5.exit");
        System.out.println("Select an option:");
        int opt = read.nextInt();
        while(opt!=5) {
            switch(opt) {
                case 1: System.out.println("len of a: "+aa.length());
                System.out.println("len of b: "+bb.length());
                break;
                case 2:
                    System.out.println("reverse of a: "+aa.reverse());
                    System.out.println("reverse of b: "+bb.reverse());
                    break;
                case 3:
                    System.out.println("enter char : ");
                    String c = read.next();

```

```

System.out.println("Enter char ");
String c2 = read.next();
System.out.println(" appended a: " + aa.append(c1));
System.out.println(" appended b: " + bb.append(c2));
break;
    
```

Case 4:

```

System.out.println("Enter char:");
String c3 = read.next();
System.out.println("Enter char:");
String c4 = read.next();
System.out.println("Index of " + c3 + " in a: " + aa.indexOf(c3));
System.out.println("Index of " + c4 + " in b: " + bb.indexOf(c4));
break;
else {
    System.out.println("wrong input");
}
    
```

```

System.out.println("Select an opt. - ")
opt = read.nextInt();
    
```



## Week 8: Packages

//File name Protection.java

```

package P1;
public class Protection {
    int n=1;
    private int n-pri = 2;
    protected int n-pro = 3;
    public int n-pub = 4;
    public Protection() {
        System.out.println("base Constructor n = " + n);
        System.out.println("n-pri = " + n-pri + " n-pro = " + n-pro);
        System.out.println("n-pub = " + n-pub);
    }
}

```

//File name Derived.java

```

package P1;
class Derived extends Protection {
    Derived() {
        System.out.println("Derived Constructor n = " + n);
        // class only
        // System.out.println("n-pri = " + n-pri);
        // System.out.println("n-pro = " + n-pro + " n-pub = " + n-pub);
    }
}

```

//File name SamePackage.java

```

package P1;
class SamePackage {
    SamePackage() {
        protection p = new Protection();
    }
}

```

```

System.out.println("Same package constructor");
System.out.println("n=" + p.n);
//System.out.print("n-pri=" + p.n-pri); // In this bound?
System.out.println("n-prv=" + p.n-prv + "\nn-pub=" + p.n-pub);
}

```

//File name Protection2.java

```

package p2;
class Protection2 extends p1.Protection{
    Protection2(){
        System.out.println("derived other package constructor");
        System.out.println("n-prv " + n-prv + "\nn-pub=" + n-pub);
    }
}

```

//File name OtherPackage.java

```

package p2;
class OtherPackage{
    OtherPackage(){
        p1.Protection p=new p1.Protection();
        System.out.println("Other Package constructor");
        System.out.println("n-pub=" + p.n-pub);
    }
}

```

//File for P1  
Package p1;

```

public class Demo{
    public static void main(String[] args){

```

```

        Protection ob1=new Protection();
    
```

```

        Derived ob2=new Derived();
    
```

```

        SamePackage ob3=new SamePackage(); 33
    
```

file for P2

package P2;

public class Demo {

public static void main(String[] args) {

Protection2 ob1 = new Protection2();

OtherPackage ob2 = new OtherPackage();

}



2 - para 6 ss 2

3 - para 6 ss 2

4 - para 6 ss 2

5 - para 6 ss 2

6 - para 6 ss 2

7 - para 6 ss 2

8 - para 6 ss 2

9 - para 6 ss 2

10 - para 6 ss 2

11 - para 6 ss 2

12 - para 6 ss 2

13 - para 6 ss 2

14 - para 6 ss 2

15 - para 6 ss 2

16 - para 6 ss 2

17 - para 6 ss 2

18 - para 6 ss 2

19 - para 6 ss 2

20 - para 6 ss 2

21 - para 6 ss 2

22 - para 6 ss 2

23 - para 6 ss 2

24 - para 6 ss 2

25 - para 6 ss 2

26 - para 6 ss 2

27 - para 6 ss 2

28 - para 6 ss 2

29 - para 6 ss 2

30 - para 6 ss 2

31 - para 6 ss 2

32 - para 6 ss 2

33 - para 6 ss 2

34 - para 6 ss 2

35 - para 6 ss 2

36 - para 6 ss 2

37 - para 6 ss 2

38 - para 6 ss 2

39 - para 6 ss 2

40 - para 6 ss 2

41 - para 6 ss 2

42 - para 6 ss 2

43 - para 6 ss 2

44 - para 6 ss 2

45 - para 6 ss 2

46 - para 6 ss 2

47 - para 6 ss 2

48 - para 6 ss 2

49 - para 6 ss 2

50 - para 6 ss 2

51 - para 6 ss 2

52 - para 6 ss 2

53 - para 6 ss 2

54 - para 6 ss 2

55 - para 6 ss 2

56 - para 6 ss 2

57 - para 6 ss 2

58 - para 6 ss 2

59 - para 6 ss 2

60 - para 6 ss 2

61 - para 6 ss 2

62 - para 6 ss 2

63 - para 6 ss 2

64 - para 6 ss 2

65 - para 6 ss 2

66 - para 6 ss 2

67 - para 6 ss 2

68 - para 6 ss 2

69 - para 6 ss 2

70 - para 6 ss 2

71 - para 6 ss 2

72 - para 6 ss 2

73 - para 6 ss 2

74 - para 6 ss 2

75 - para 6 ss 2

76 - para 6 ss 2

77 - para 6 ss 2

78 - para 6 ss 2

79 - para 6 ss 2

80 - para 6 ss 2

81 - para 6 ss 2

82 - para 6 ss 2

83 - para 6 ss 2

84 - para 6 ss 2

85 - para 6 ss 2

86 - para 6 ss 2

87 - para 6 ss 2

88 - para 6 ss 2

89 - para 6 ss 2

90 - para 6 ss 2

91 - para 6 ss 2

92 - para 6 ss 2

93 - para 6 ss 2

94 - para 6 ss 2

95 - para 6 ss 2

96 - para 6 ss 2

97 - para 6 ss 2

98 - para 6 ss 2

99 - para 6 ss 2

100 - para 6 ss 2

101 - para 6 ss 2

102 - para 6 ss 2

103 - para 6 ss 2

104 - para 6 ss 2

105 - para 6 ss 2

106 - para 6 ss 2

107 - para 6 ss 2

108 - para 6 ss 2

109 - para 6 ss 2

110 - para 6 ss 2

111 - para 6 ss 2

112 - para 6 ss 2

113 - para 6 ss 2

114 - para 6 ss 2

115 - para 6 ss 2

116 - para 6 ss 2

117 - para 6 ss 2

118 - para 6 ss 2

119 - para 6 ss 2

120 - para 6 ss 2

121 - para 6 ss 2

122 - para 6 ss 2

123 - para 6 ss 2

124 - para 6 ss 2

125 - para 6 ss 2

126 - para 6 ss 2

127 - para 6 ss 2

128 - para 6 ss 2

129 - para 6 ss 2

130 - para 6 ss 2

131 - para 6 ss 2

132 - para 6 ss 2

133 - para 6 ss 2

134 - para 6 ss 2

135 - para 6 ss 2

136 - para 6 ss 2

137 - para 6 ss 2

138 - para 6 ss 2

139 - para 6 ss 2

140 - para 6 ss 2

141 - para 6 ss 2

142 - para 6 ss 2

143 - para 6 ss 2

144 - para 6 ss 2

145 - para 6 ss 2

146 - para 6 ss 2

147 - para 6 ss 2

148 - para 6 ss 2

149 - para 6 ss 2

150 - para 6 ss 2

151 - para 6 ss 2

152 - para 6 ss 2

153 - para 6 ss 2

154 - para 6 ss 2

155 - para 6 ss 2

156 - para 6 ss 2

157 - para 6 ss 2

158 - para 6 ss 2

159 - para 6 ss 2

160 - para 6 ss 2

161 - para 6 ss 2

162 - para 6 ss 2

163 - para 6 ss 2

164 - para 6 ss 2

165 - para 6 ss 2

166 - para 6 ss 2

167 - para 6 ss 2

168 - para 6 ss 2

169 - para 6 ss 2

170 - para 6 ss 2

171 - para 6 ss 2

172 - para 6 ss 2

173 - para 6 ss 2

174 - para 6 ss 2

175 - para 6 ss 2

176 - para 6 ss 2

177 - para 6 ss 2

178 - para 6 ss 2

179 - para 6 ss 2

180 - para 6 ss 2

181 - para 6 ss 2

182 - para 6 ss 2

183 - para 6 ss 2

184 - para 6 ss 2

185 - para 6 ss 2

186 - para 6 ss 2

187 - para 6 ss 2

188 - para 6 ss 2

189 - para 6 ss 2

190 - para 6 ss 2

191 - para 6 ss 2

192 - para 6 ss 2

193 - para 6 ss 2

194 - para 6 ss 2

195 - para 6 ss 2

196 - para 6 ss 2

197 - para 6 ss 2

198 - para 6 ss 2

199 - para 6 ss 2

200 - para 6 ss 2

201 - para 6 ss 2

202 - para 6 ss 2

203 - para 6 ss 2

204 - para 6 ss 2

205 - para 6 ss 2

206 - para 6 ss 2

207 - para 6 ss 2

208 - para 6 ss 2

209 - para 6 ss 2

210 - para 6 ss 2

211 - para 6 ss 2

212 - para 6 ss 2

213 - para 6 ss 2

214 - para 6 ss 2

215 - para 6 ss 2

216 - para 6 ss 2

217 - para 6 ss 2

218 - para 6 ss 2

219 - para 6 ss 2

220 - para 6 ss 2

221 - para 6 ss 2

222 - para 6 ss 2

223 - para 6 ss 2

224 - para 6 ss 2

225 - para 6 ss 2

226 - para 6 ss 2

227 - para 6 ss 2

228 - para 6 ss 2

229 - para 6 ss 2

230 - para 6 ss 2

231 - para 6 ss 2

232 - para 6 ss 2

233 - para 6 ss 2

234 - para 6 ss 2

235 - para 6 ss 2

236 - para 6 ss 2

237 - para 6 ss 2

238 - para 6 ss 2

239 - para 6 ss 2

240 - para 6 ss 2

241 - para 6 ss 2

242 - para 6 ss 2

243 - para 6 ss 2

244 - para 6 ss 2

245 - para 6 ss 2

246 - para 6 ss 2

247 - para 6 ss 2

248 - para 6 ss 2

249 - para 6 ss 2

250 - para 6 ss 2

251 - para 6 ss 2

252 - para 6 ss 2

253 - para 6 ss 2

254 - para 6 ss 2

255 - para 6 ss 2

256 - para 6 ss 2

257 - para 6 ss 2

258 - para 6 ss 2

259 - para 6 ss 2

260 - para 6 ss 2

261 - para 6 ss 2

262 - para 6 ss 2

263 - para 6 ss 2

264 - para 6 ss 2

265 - para 6 ss 2

266 - para 6 ss 2

## Week 9: Exception Handling

```
import java.util.*;
class Main{
    public static void main (String[] args){
        int j;
        Scanner i = new Scanner (System.in);
        System.out.println ("Size of array : ");
        j = i.nextInt();
        try {
            int arr[] = new int[j];
            System.out.println ("size of array is : " + j);
            if (arr.length > 0)
                System.out.println ("Size of array is positive");
        } catch (NegativeArraySizeException e) {
            System.out.println ("After try block ");
            System.out.println (e);
            System.out.println ("Exception occurred ");
        }
    }
}
```

BY

Week 10: Write a Java program by multi-thread consists of three threads for random, even, odd

class Random\_num extends Thread {

String nm;

Random\_num(String name) {

nm = name;

Thread t = new Thread(this, nm);

t.start();

}

public void run() {

try {

for (int i=0; i<5; i++) {

double r = Math.random() \* 10;

int rand = (int) r;

System.out.println("Random number : " + rand);

if (rand % 2 == 0)

new even\_sq(rand, "even");

else

new odd\_cd(rand, "odd");

Thread.sleep(1000);

}

}

catch (InterruptedException e) {

System.out.println("Error : " + e);

}

}

class even\_sq extends Thread {

int n;

String nm;

even\_sq(int rand, String name) {

nm = name;

n = rand;

```

Thread t=new Thread(this.n1);
t.start();
}

public void run() {
    System.out.println("Square of " + n + " = " + (n*n));
}

class odd_cd extends Thread {
    int n;
    String n2;
    odd_cd(int rand, String name) {
        n2=name;
        n=rand;
        Thread t=new Thread(this.n2);
        t.start();
    }

    public void run() {
        System.out.println("Cube of " + n + " = " + (n*n*n));
    }
}

public class odd_eve_Thread {
    public static void main(String[] args) {
        new Rand_num("prog");
    }
}

```

(B)

Write a Java program for passing parameters to Applets.

```

import java.applet.*;
import java.awt.*;
import java.io.*;

<applet code="ParameterPro" width=300 height=300>
<param name="name" value="lugman">
<param name="year" value="first year">
<param name="branch" value="CSE">
<param name="College" value="RVR & JC">
<param name="details" value="Student Details">
<param name="Fee" value="70000">
</applet>*/
```

```

public class ParameterPro extends Applet {
    public void paint(Graphics g) {
        g.drawString(getParameter("details"), 150, 20);
        g.drawString("Name: " + getParameter("name"), 50, 50);
        g.drawString("Branch: " + getParameter("branch"), 50, 70);
        g.drawString("Year: " + getParameter("year"), 50, 90);
        g.drawString("College: " + getParameter("College"), 50, 110);
        int d = Integer.parseInt(getParameter("Fee"));
        g.drawString("Fee: " + d, 50, 130);
    }
}
```

✓

Write a Java program to implement Graphics class and color class.

```

import java.awt.*;
import java.applet.*;

<applet code="ColorDemo" width=300 height=200>
</applet>

public class ColorDemo extends Applet {
    public void paint(Graphics g) {
        g.setColor(Color.black);
        g.drawString("Java program", 50, 50);
        g.drawLine(20, 30, 20, 300);
        g.drawRect(70, 100, 30, 30);
        g.fillRect(170, 100, 30, 30);
        g.drawOval(70, 200, 30, 30);
        g.fillOval(170, 200, 30, 30);
        g.drawArc(90, 150, 30, 30, 30, 270);
        g.fillArc(290, 150, 30, 30, 0, 180);
        g.drawRoundRect(190, 10, 60, 50, 15, 15);

        int xpoints[] = {30, 40, 30, 40, 30};
        int ypoints[] = {30, 30, 40, 40, 30};
        int num = 5;
        g.drawPolygon(xpoints, ypoints, num);
        set Background(Color.white);
    }
}

```

y

B

Week-18.

Write a Java program for handling mouse events

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*applet code="MouseEvents" width=300 height=100&gt;

&lt;applet&gt;

/ public class MouseEvents extends Applet implements MouseListener,

MouseMotionListener {

String msg = "";

int mouseX = 0, mouseY = 0;

public void init() {

addMouseListener(this);

addMouseMotionListener(this);

{

public void mouseClicked(MouseEvent me) {

mouseX = 0;

mouseY = 10;

msg = "Mouse clicked. ";

repaint();

{

public void mouseEntered(MouseEvent me) {

mouseX = 0;

mouseY = 10;

msg = "Mouse entered. ";

repaint();

{

public void mouseExited(MouseEvent me) {

mouseX = 0;

mouseY = 10;

msg = "Mouse exited. ";

repaint(); }

```
public void mousePressed(MouseEvent me) {
    mouseX = me.getX();
    mouseY = me.getY();
    msg = "Down";
    repaint();
}

public void mouseReleased(MouseEvent me) {
    mouseX = me.getX();
    mouseY = me.getY();
    msg = "Up";
    repaint();
}

public void mouseDragged(MouseEvent me) {
    mouseX = me.getX();
    mouseY = me.getY();
    msg = "*";
    ShowStatus("Dragging mouse at " + mouseX + ", " + mouseY);
    repaint();
}

public void mouseMoved(MouseEvent me) {
    ShowStatus("Moving mouse at " + me.getX() + ", " + me.getY());
}

public void paint(Graphics g) {
    g.drawString(msg, mouseX, mouseY);
}
```

✓  
82

Write a java program for handling key events

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

\*<applet code="Simplekey" width=300 height=100>

</applet>\*>

public class Simplekey extends Applet implements KeyListener {

String msg = "";

int x = 10, y = 20;

public void init() {

addKeyListener(this);

}

public void keyPressed(KeyEvent ke) {

showStatus("Key Down");

y

public void keyReleased(KeyEvent ke) {

showStatus("key Up");

y

public void keyTyped(KeyEvent ke) {

msg += ke.getKeyChar();

repaint();

y

public void paint(Graphics g) {

g.drawString(msg, x, y);

y

3

 ✓

write java program to implement AWT Components.  
 applet code = "program13"  
 applet> width = 600 height = 600>

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class program13 extends Applet implements ActionListener,
```

```
button b1, b2, b3;
```

```
choice m1;
```

```
checkbox mfm, asec, bsec, csec;
```

```
extfield tf2, tf1, tf3;
```

```
label l1, l2, l3;
```

```
checkboxGroup cbg;
```

```
public void init()
```

```
  cbg = new CheckboxGroup();
```

```
  l1 = new Label("Welcome to RVRJCE");
```

```
  l1.setBounds(100, 10, 150, 50);
```

```
  l2 = new Label("Branch Name:");
```

```
  l2.setBounds(0, 75, 80, 30);
```

```
  tf1 = new Textfield();
```

```
  tf1.setBounds(140, 75, 100, 20);
```

```
  l3 = new Label("Registered Number:");
```

```
  l3.setBounds(0, 100, 130, 30);
```

```
  asec = new Checkbox("A-Sec", cbg, true);
```

```
  asec.setBounds(300, 75, 50, 30);
```

```
bsec = new Checkbox ("B-Sec", cbg, true);
```

```
bsec.setBounds (360, 75, 130, 30);
```

```
csec = new Checkbox ("C-Sec", cbg, true);
```

```
csec.setBounds (300, 100, 130, 30);
```

```
tf2 = new TextField();
```

```
tf2.setBounds (140, 100, 100, 20);
```

```
m = new Checkbox ("male", true);
```

```
m.setBounds (20, 140, 80, 30);
```

```
fm = new Checkbox ("female", true);
```

```
fm.setBounds (100, 140, 80, 30);
```

```
b1 = new Button ("Hostel");
```

```
b1.setBounds (20, 200, 60, 30);
```

```
b2 = new Button ("Day-scholar");
```

```
b2.setBounds (90, 200, 90, 30);
```

```
b3 = new Button ("clear");
```

```
b3.setBounds (190, 200, 60, 30);
```

```
tf3 = new TextField();
```

```
tf3.setBounds (30, 270, 150, 20);
```

```
ml = new Choice();
```

```
ml.setBounds (190, 140, 80, 80);
```

```
ml.add ("telugu");
```

```
ml.add ("hindi");
```

```
ml.add ("English");
```

```
add (asec);
```

```
add (bsec);
```

```
add (csec);
```

```
add (ml);
```

```

add(lb1);
add(lb2);
add(lb3);
add(tf1);
add(tf2);
add(tf3);
add(m);
add(fm);
add(b1);
add(b2);
add(b3);
m.addItemListener(this);
fm.addItemListener(this);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
m.addItemListener(this);
aSec.addItemListener(this);
bSec.addItemListener(this);
cSec.addItemListener(this);
setLayout(null);

3 public void itemStateChanged(ItemEvent e){
    repaint();
}

3 public void actionPerformed(ActionEvent e){
    if(e.getSource() == b1)
        tf3.setText("Counter 1");
    if(e.getSource() == b2)
        tf3.setText("Counter 2");
}

```

```
if (e.getSource() == b3) {  
    tf2.setText("");  
    tf1.setText("");  
    tf3.setText("");  
}
```



Write a java program to develop SWING Components.

```

import java.awt.FlowLayout;
import java.swing.*;
public class JFrameExample {
    public static void main(String args[]) {
        JFrame frame = new JFrame("JFrame Example");
        JPanel panel = new JPanel();
        panel.setLayout(new FlowLayout());
        JLabel label = new JLabel("Swing frame Example");
        JButton button = new JButton();
        button.setText("Button");
        JCheckBox cb = new JCheckBox("C++");
        String country[] = {"India", "AUS", "USA", "England"};
        JComboBox cb = new JComboBox(country);
        cb.setBounds(50, 50, 90, 20);
        JScrollPane sb = new JScrollPane();
        sb.setBounds(100, 100, 50, 100);
        JRadioButton r1 = new JRadioButton("Male");
        r1.setBounds(75, 50, 100, 30);
        String data[][] = {{"101", "Amit", "670000"}, {"102", "Jai", "78000"}, {"101", "Gachin", "70000"}};
        String column [] = {"ID", "Name", "Salary"};
        JTable jt = new JTable(data, column);
        jt.setBounds(30, 40, 200, 300);
        panel.add(label);
        panel.add(button);
    }
}

```

```
panel.add(checkBox1);  
panel.add(cb);  
panel.add(sb);  
panel.add(r1);  
panel.add(jt);  
frame.add(panel);  
frame.setSize(200, 300);  
frame.setLocationRelativeTo(null);  
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
frame.setVisible(true);
```

2

3

BB

Write a java program to find maximum and minimum values from given type of elements using generics.

```

class MyClass<T extends Comparable<T>>{
    T[] vals;
    MyClass(T[] o){
        vals=o;
    }
    public T min(){
        T v=vals[0];
        for(int i=1; i<vals.length; i++)
            if(vals[i].compareTo(v)<0)
                v=vals[i];
        return v;
    }
    public T max(){
        T v=vals[0];
        for(int i=1; i<vals.length; i++)
            if(vals[i].compareTo(v)>0)
                v=vals[i];
        return v;
    }
}
class genDemo{
    public static void main(String args[]){
        int i;
        Integer inums[]={10, 2, 5, 4, 6, 19};
        Character chs[]={'v', 'P', 'S', 'a', 'n', 'h'};
    }
}

```

Double d[] = {20.2, 45.4, 71.6, 88.3, 54.6, 10.4};

MyClass<Integer> iob = new MyClass<Integer>(inums);

MyClass<Character> cob = new MyClass<Character>(chs);

MyClass<Double> dob = new MyClass<Double>(d);

System.out.println("Max in inums: " + iob.max());

System.out.println("Min in inums: " + iob.min());

System.out.println("Max in chs: " + cob.max());

System.out.println("Min in chs: " + cob.min());

System.out.println("Max in chs: " + dob.max());

System.out.println("Min in chs: " + dob.min());

J

3

✓  
BB