

1 . Java Basics

- (a) Write a Java program that takes a number as input and prints its multiplication table upto 10.**

```
import java.util.Scanner;
public class demo
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Input a number: ");
        int num1 = in.nextInt();
        for (inti=0; i< 10; i++)
        {
            System.out.println(num1 + " x " + (i+1) + " = " + (num1 * (i+1)));
        }
    }
}
```

- (b) Write a Java program to display the following pattern.**

**

*

```
class star
{
    public static void main(String[] args)
    {
        inti, j;
        for(i=5; i>=1; i--)
        {
            for(j=1; j<=i; j++)
            {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

- (c) Write a Java program to print the area and perimeter of a circle.**

```
public class area
{
    private static final double radius = 7.5;
    public static void main(String[] args)
    {
        double perimeter = 2 * Math.PI * radius;
        double area = Math.PI * radius * radius;
```

```
System.out.println("Perimeter is = " + perimeter);
System.out.println("Area is = " + area);
    }
}
```

2. Use of Operators

(a) Write a Java program to add two binary numbers.

```
import java.util.Scanner;
public class binary
{
    public static void main(String[] args)
    {
        long binary1, binary2;
        inti = 0, remainder = 0;
        int[] sum = new int[20];
        Scanner in = new Scanner(System.in);
        System.out.print("Input first binary number: ");
        binary1 = in.nextLong();
        System.out.print("Input second binary number: ");
        binary2 = in.nextLong();
        while (binary1 != 0 || binary2 != 0)
        {
            sum[i++] = (int)((binary1 % 10 + binary2 % 10 + remainder) % 2);
            remainder = (int)((binary1 % 10 + binary2 % 10 + remainder) / 2);
            binary1 = binary1 / 10;
            binary2 = binary2 / 10;
        }
        if (remainder != 0)
        {
            sum[i++] = remainder;
        }
        --i;
        System.out.print("Sum of two binary numbers: ");
        while (i >= 0)
        {
            System.out.print(sum[i--]);
        }
        System.out.print("\n");
    }
}
```

(b) Write a Java program to convert a decimal number to binary number and vice versa.

```
class DtoB
{
    public static void main(String arg[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a decimal number");
    }
}
```

```

    int n=sc.nextInt();
    System.out.print("Binary number is :");
    binary(n);

}
static void binary(intnum)
{
    inti = 0;
    int bin[]=new int[100];
    bin[0]=0;
    while(num>0)
    {
        bin[i++] = num%2;
        num = num/2;
    }
    for(int j =i-1;j >= 0;j--)
    {
        System.out.print(bin[j]);
    }
}
}

```

(c) Write a Java program to reverse a string.

```

import java.util.*;
class ReverseString
{
    public static void main(String args[])
    {
        String original, reverse = "";
        Scanner in = new Scanner(System.in);
        System.out.println("Enter a string to reverse");
        original = in.nextLine();
        int length = original.length();
        for ( inti = length - 1 ; i>= 0 ; i-- )
            reverse = reverse + original.charAt(i);
        System.out.println("Reverse of entered string is: "+reverse);
    }
}

```

3. Java Data Types**(a) Write a Java program to count the letters, spaces, numbers and other characters of an input string.**

```

import java.util.Scanner;
public class cout{
    public static void main(String[] args) {
        String test = "my name is dhirajkumar sinha.mob-7506669926?";
        count(test);
    }
}

```

```

public static void count(String x){
    char[] ch = x.toCharArray();
    int letter = 0;
    int space = 0;
    int num = 0;
    int other = 0;
    for(int i = 0; i < x.length(); i++){
        if(Character.isLetter(ch[i])){
            letter ++;
        }
        else if(Character.isDigit(ch[i])){
            num ++;
        }
        else if(Character.isSpaceChar(ch[i])){
            space ++;
        }
        else{
            other ++;
        }
    }
    System.out.println("The string is : Aakiu, I swdskieo 236587. GH kiu: sieo??
25.33");

    System.out.println("letter: " + letter);
    System.out.println("space: " + space);
    System.out.println("number: " + num);
    System.out.println("other: " + other);
}

```

(b) Implement a Java function that calculates the sum of digits for a given char array consisting of the digits '0' to '9'. The function should return the digit sum as a long value.

```

import java.util.Scanner;
public class digit {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Input an integer: ");
        long n = input.nextLong();
        System.out.println("The sum of the digits is: " + sumDigits(n));
    }
    public static int sumDigits(long n) {
        int sum = 0;
        while (n != 0) {
            sum += n % 10;
            n /= 10;
        }
        return sum;
    }
}

```

}

(c) Find the smallest and largest element from the array .

```

public class LargeSmall{
    public static void main(String[] args) {
        //array of 10 numbers
        int numbers[] = new int[]{32,43,53,54,32,65,63,98,43,23};
        //assign first element of an array to largest and smallest
        int smallest = numbers[0];
        int largest = numbers[0];
        for(int i=1; i<numbers.length; i++)
        {
            if(numbers[i] > largest)
                largest = numbers[i];
            else if (numbers[i] < smallest)
                smallest = numbers[i];
        }
        System.out.println("Largest Number is : " + largest);
        System.out.println("Smallest Number is : " + smallest);
    }
}

```

4. Methods and Constructors**(a) Designed a class SortData that contains the method asc() and desc().**

```

import java.util.*;
class SortData
{
    Scanner input = new Scanner(System.in);
    int num, i;
    int arr[];
    int temp = 0;
    public void getData()
    {
        System.out.print("Enter the size of array: ");
        num = input.nextInt();
        arr = new int[num];
        System.out.print("Enter the number: ");
        for(i=0; i<num; i++)
        {
            arr[i] = input.nextInt();
        }
    }
    void putData()
    {
        System.out.print("Given numbers are: ");
        for(i=0; i<num; i++)
        {
            System.out.println(arr[i]);
        }
    }
}

```

```
void asce()
{
    for(i=0;i<num;i++)
    {
        for(int j=i+1;j<num;j++)
        {
            if(arr[i]>arr[j])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
    System.out.print("Ascending order of number are: ");
    for(int i=0;i<num;i++)
    {
        System.out.println(arr[i]);
    }
}

void desc()
{
    for(i=0;i<num;i++)
    {
        for(int j=i+1;j<num;j++)
        {
            if(arr[i]<arr[j])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
    System.out.print("Descending order of number are: ");
    for(int i=0;i<num;i++)
    {
        System.out.println(arr[i]);
    }
}

public static void main(String args[])
{
    sortdata ob=new sortdata();
    ob.getdata();
    ob.putdata();
    ob.asce();
}
```

```
ob.desc();  
}  
}
```

(b) Designed a class that demonstrates the use of constructor and destructor.

```
import java.io.*;
```

```
class A
```

```
{  
  
    int i;  
  
    public A(int i)  
    {  
        this.i=i;  
    }  
  
    protected void finalize() throws Throwable  
    {  
        System.out.println("From finalize method, i="+i);  
        super.finalize();  
    }  
}
```

```
public class construct
```

```
{  
  
    public static void main(String args[])  
    {  
        A a1=new A(10);  
        A a2=new A(20);  
  
        try  
        {  
            a1.finalize();  
        }  
  
        catch(Throwable e)  
        {  

```

```
        e.printStackTrace();
    }
    a1=a2;
    System.out.println("sucessfully");
}
}
```

(c) Write a java program to demonstrate the implementation of abstract class.

```
abstract class mal{
    //abstract method
    public abstract void sound();
}
//Dog class extends mal class
public class Dog extends mal{
    public void sound(){
        System.out.println("Woof");
    }
    public static void main(String args[]){
        malobj = new Dog();
        obj.sound();
    }
}
```

5. Inheritance

(a)Write a java program to implement single level inheritance.

```
class sin{
    static int num1=10;
    static int num2=5;
}
class man extends sin{
    public static void main(String[] args){
        int num3=2;
        int result=num1+num2+num3;
        System.out.println("Result of child class is "+result);
    }
}
```

(b) Write a java program to implement method overriding

```
class Human{
    //Overridden method
    public void eat()
    {
        System.out.println("Human is eating");
    }
}
class Boy extends Human{
```



```
//Overriding method
public void eat(){
    System.out.println("Boy is eating");
}
public static void main( String args[]) {
    Boy obj = new Boy();
    //This will call the child class version of eat()
    obj.eat();
}
}
```

(c) Write a java program to implement multiple inheritance.

```
interface PI1
{
    // default method
    default void show()
    {
        System.out.println("Default PI1");
    }
}
interface PI2
{
    // Default method
    default void show()
    {
        System.out.println("Default PI2");
    }
}
// Implementation class code
classTestClass implements PI1, PI2
{
    // Overriding default show method
    public void show()
    {
        // use super keyword to call the show
        // method of PI1 interface
        PI1.super.show();
        // use super keyword to call the show
        // method of PI2 interface
        PI2.super.show();
    }
    public static void main(String args[])
    {
        TestClass d = new TestClass();
        d.show();
    }
}
```

```
}
```

6. Packages and Arrays

(a)Create a package, Add the necessary classes and import the package in java class.

```
//folder myPackage
```

```
packagemyPackage;
```

```
public class MyClass
```

```
{
```

```
public void getNames(String s)
```

```
{
```

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

```
}
```

```
}
```

```
importmyPackage.MyClass;
```

```
public class PrintName
```

```
{
```

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

```
{
```

```
    // Initializing the String variable
```

```
    // with a value
```

```
    String name = "GeeksforGeeks";
```

```
    // Creating an instance of class MyClass in
```

```
    // the package.
```

```
MyClassobj = new MyClass();
```

```
obj.getNames(name);
```

```
}
```

```
}
```

(b) Write a java program to add two matrices and print the resultant matrix

```
importjava.util.Scanner;
```

```
classAddTwoMatrix
```

```
{
```

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

```
{
```

```
int m, n, c, d;
```

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

```
System.out.println("Enter the number of rows and columns of  
matrix");
```

```
m = in.nextInt();
```

```
n =in.nextInt();
```

```
int first[][] = new int[m][n];
```

```
int second[][] = new int[m][n];
```

```
int sum[][] = new int[m][n];
```

```
System.out.println("Enter the elements of first matrix");
```

```
for ( c = 0 ; c < m ; c++ )
```

```
for ( d = 0 ; d < n ; d++ )
```

```
first[c][d] = in.nextInt();
```

```
System.out.println("Enter the elements of second matrix");
```

```

for ( c = 0 ; c < m ; c++ )
for ( d = 0 ; d < n ; d++ )
second[c][d] = in.nextInt();
for ( c = 0 ; c < m ; c++ )
for ( d = 0 ; d < n ; d++ )
sum[c][d] = first[c][d] + second[c][d]; //replace '+' with '-' to
subtract matrices
System.out.println("Sum of entered matrices:-");
for ( c = 0 ; c < m ; c++ )
{
for ( d = 0 ; d < n ; d++ )
System.out.print(sum[c][d]+"\\t");
System.out.println();
}
}
}

```

(c) Write a java program for multiplying two matrices and print the product for the same.

```

public class MatrixMultiplicationExample{
public static void main(String args[]){
//creating two matrices
int a[][]={{1,1,1},{2,2,2},{3,3,3}};
int b[][]={{1,1,1},{2,2,2},{3,3,3}};
//creating another matrix to store the multiplication of two matrices
int c[][]=new int[3][3]; //3 rows and 3 columns
//multiplying and printing multiplication of 2 matrices
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
c[i][j]=0;
for(int k=0;k<3;k++){
{
c[i][j]+=a[i][k]*b[k][j];
}
}
}
}
}
}
}
}
}

```

7. Vectors and Multithreading

(a) Write a java program to implement the vectors.

```

import java.util.ArrayList;
import java.util.Scanner;
public class Vector<T>
{
private int capacity;
private int size;

```

```
private ArrayList<T> vector;
private static final int INCREMENT_FACTOR = 5;
public Vector(int size)
{
    this.size = size;
    this.capacity = size + INCREMENT_FACTOR;
    vector = new ArrayList<T>();
}
public void store(int index, T value)
{
    try
    {
        vector.set(index, value);
    } catch (IndexOutOfBoundsException indexOutOfBounds)
    {
        if (index >= 0 && (index < size))
        {
            vector.add(index, value);
        }
        if (index >= 0 && (index >= size && index < capacity))
        {
            vector.add(index, value);
            size = index + 1;
            if (size == capacity)
                capacity = capacity + INCREMENT_FACTOR;
        }
        if (index >= capacity)
        {
            throw new IndexOutOfBoundsException();
        }
    }
}
public T get(int index)
{
    try
    {
        return vector.get(index);
    } catch (IndexOutOfBoundsException indexOutOfBounds)
    {
        return null;
    }
}
public int getSize()
{
    return size;
}
```

```

public int getCapacity()
{
    return capacity;
}
public static void main(String... arg)
{
    int size;
    int num;
    int value;
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter the initial size of the vector");
    size = scanner.nextInt();
    Vector<Integer> vector = new Vector<>(size);
    System.out.println("Enter the number of elements ");
    num = scanner.nextInt();
    System.out.println("Enter the values");
    for (int index = 0; index < num; index++)
    {
        value = scanner.nextInt();
        vector.store(index, value);
    }
    System.out.println("The Entered Values are");
    for (int index = 0; index < vector.getSize(); index++)
    {
        System.out.print(vector.get(index) + "\t");
    }
    System.out.println("\nTHE SIZE OF THE VECTOR IS " +
        vector.getSize());
    System.out.println("THE CAPACITY OF THE VECTOR IS " +
        vector.getCapacity());
    scanner.close();
}
}

```

(b) Write a java program to implement thread life cycle.

A.java

```

public class A extends Thread {
    public void run()
    {
        System.out.println("Thread A");
        System.out.println("i in Thread A ");
        for(int i=1;i<=5;i++)
        {
            System.out.println("i = " + i);
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {

```

```
// TODO Auto-generated catch block
e.printStackTrace();
}
}
System.out.println("Thread A Completed.");
}
}
```

B.java

```
public class B extends Thread {
    public void run()
    {
        System.out.println("Thread B");
        System.out.println("i in Thread B ");
        for(int i=1;i<=5;i++)
        {
            System.out.println("i = " + i);
        }
        System.out.println("Thread B Completed.");
    }
}
```

Main.java

```
public class Main {
    public static void main(String[] args) {
        //life cycle of Thread
        // Thread's New State
        A threadA = new A();
        B threadB = new B();
        // Both the above threads are in runnable state
        //Running state of thread A & B
        threadA.start();
        //Move control to another thread
        threadA.yield();
        //Blocked State of thread B
        try {
            threadA.sleep(1000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        threadB.start();
        System.out.println("Main Thread End");
    }
}
```

(c) Write a java program to implement multithreading.

```
// Java code for thread creation by implementing
// the Runnable Interface
```

```

class MultithreadingDemo implements Runnable
{
    public void run()
    {
        try
        {
            // Displaying the thread that is running
            System.out.println ("Thread " + Thread.currentThread().getId() + " is
running");
        }
        catch (Exception e)
        {
            // Throwing an exception
            System.out.println ("Exception is caught");
        }
    }
}
// Main Class
class Multithread
{
    public static void main(String[] args)
    {
        int n = 8; // Number of threads
        for (inti=0; i<8; i++)
        {
            Thread object = new Thread(new MultithreadingDemo());
            object.start();
        }
    }
}

```

8. File Handling**(a) Write a java program to open a file and display the contents in the console window.**

```

import java.util.Scanner;
import java.io.*;
public class WritingTextFiles{
    public static void main (String [] args) throws IOException{
        FileWriter fw= new FileWriter("testing.txt");
        Scanner in= new Scanner (System.in);
        String testwords=in.nextLine();
        fw.write(testwords);
        BufferedReader r = new BufferedReader( new FileReader(
"testing.txt" ) );
        System.out.print(r);
        fw.close();
    }
}

```

(b) Write a java program to copy the contents from one file to other file.

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
public class CopyExample
{
    public static void main(String[] args)
    {
        FileInputStream instream = null;
        FileOutputStream outstream = null;
        try{
            File infile = new File("d:\\testing.txt");
            File outfile = new File("C:\\MyOutputFile.txt");
            instream = new FileInputStream(infile);
            outstream = new FileOutputStream(outfile);
            byte[] buffer = new byte[1024];
            int length;
            /*copying the contents from input stream to
             * output stream using read and write methods
             */
            while ((length = instream.read(buffer)) > 0){
                outstream.write(buffer, 0, length);
            }
            //Closing the input/output file streams
            instream.close();
            outstream.close();
            System.out.println("File copied successfully!!");
        }catch(IOException ioe){
            ioe.printStackTrace();
        }
    }
}
```

(c) Write a java program to read the student data from user and store it in the file.

```
import java.io.File;
public class FileClassMethods {
    public static void main(String[] args) {
        File myFile = new File("Data.txt");
        if (myFile.exists()) {
            System.out.println(myFile.getName() + " exists");
            System.out.println("The file is " + myFile.length() + " bytes long");
            if (myFile.canRead())
                System.out.println(" ok to read");
            else
                System.out.println(" not ok to read");
            if (myFile.canWrite())
```



```
        System.out.println(" ok to write");
    else
        System.out.println(" not ok to write");
    System.out.println("path: " +myFile.getAbsolutePath());
    System.out.println("File Name: "+ myFile.getName());
    System.out.println("File Size: "+ myFile.length() + " bytes");
} else
    System.out.println("File does not exist");
}
}
```

9. GUI and Exception Handling

(a) Design a AWT program to print the factorial for an input value.

```
import java.awt.*;
import java.awt.event.*;
class ex2 extends Frame implements ActionListener
{
    Label l1,l2,l3;
    TextField t1;
    Button b1,b2;
    ex2()
    {
        super("Factorial of a number");
        setLayout(null);
        setSize(500,400);
        setVisible(true);
        b1=new Button("Find");
        b2=new Button("Clear");
        l1=new Label("Enter the number:");
        l2=new Label("The factorial is:");
        l3=new Label(null);
        t1=new TextField();
        l1.setBounds(100,50,120,20);
        add(l1);
        t1.setBounds(240,50,50,20);
        add(t1);
        l2.setBounds(100,80,130,20);
        add(l2);
        l3.setBounds(240,80,50,20);
        add(l3);
        b1.setBounds(200,150,50,20);
        add(b1);
        b2.setBounds(270,150,50,20);
        add(b2);
        b1.addActionListener(this);
        b2.addActionListener(this);
        Toolkit t=getToolkit();
```

```

        Dimension d=t.getScreenSize();
int h=(int) d.getHeight();
int w=(int) d.getWidth();
setLocation(w/4, h/4);
addWindowListener(new WindowAdapter()
{
public void windowClosing(WindowEvent we)
{
System.exit(0);
}
});
}
public void actionPerformed(ActionEvent ae)
{
int n;
if(ae.getSource()==b1)
{
n=Integer.parseInt(t1.getText().trim());
l3.setText(Integer.toString(fact(n)));
}
else
{
t1.setText(null);
l3.setText(null);
t1.requestFocus();
}
}
int fact(int n)
{
int f=1;
for(int i=1;i<=n;i++)
{
f=f*i;
}
return f;
}
public static void main(String s[])
{
ex2ob=new ex2();
}
}

```

(b) Design an AWT program to perform various string operations like reverse string, string concatenation etc.

```
public class concat
```

```
{
```

```
public static void main(String args[])
{
    String str1="hello";
    String str2="Word";
    String str3=str1+str2;
    System.out.println("String concat using+ operator:"+str3);
    String str4=str1.concat(str2);
    System.out.println("String using concat String + concat method:"+str4);
}
}
```

(c) Write a java program to implement exception handling.

```
class Division {
public static void main(String[] args) {
int a, b, result;
    Scanner input = new Scanner(System.in);
    System.out.println("Input two integers");
    a = input.nextInt();
    b = input.nextInt();
    // try block
    try {
    result = a / b;
    System.out.println("Result = " + result);
    }
    // catch block
    catch (ArithmeticException e) {
    System.out.println("Exception caught: Division by zero.");
    }
    }
}
```

10. GUI Programming.

(a) Design an AWT application that contains the interface to add student information and display the same

```
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
class Frame1 extends Frame implements ActionListener
{
    String msg="";
    Button btnNew,btnSubmit,btnView;
    Label lblName,lblAge,lblAddr,lblGender,lblQua;
```

```
TextField txtName, txtAge;
TextArea txtAddr, txtAns;
CheckboxGroup ChkGrp;
    Checkbox chkMale, chkFemale;
    Checkbox chkMca, chkBca, chkBba, chkMba;
Frame1(String name)
{
    super(name);
    setLayout(new GridLayout(3,2));
    lblName = new Label("Name: ");
    lblAge = new Label("Age: ");
    lblAddr = new Label("Address : ");
    lblGender = new Label("Gender: ");
    lblQua = new Label("Qualification: ");
    txtName = new TextField(20);
    txtAge = new TextField(20);
    txtAddr = new TextArea();
    ChkGrp = new CheckboxGroup();
    chkMale = new Checkbox("Male", ChkGrp, false);
    chkFemale = new Checkbox("Female", ChkGrp, false);
    chkMca = new Checkbox("MCA");
    chkBca = new Checkbox("BCA");
    chkMba = new Checkbox("MBA");
    chkBba = new Checkbox("BBA");
    btnNew = new Button("NEW");
    btnSubmit = new Button("SUBMIT");
    btnView = new Button("VIEW");
    btnNew.addActionListener(this);
    btnSubmit.addActionListener(this);
    btnView.addActionListener(this);
    add(lblName);
    add(txtName);
    add(lblAge);
    add(txtAge);
    add(lblAddr);
    add(txtAddr);
    add(lblGender);
    add(chkMale);
    add(chkFemale);
    add(lblQua);
    add(chkBca);
    add(chkBba);
    add(chkMca);
    add(chkMba);
    add(btnNew);
    add(btnSubmit);
```

```
add(btnView);
txtAns = new TextArea();
add(txtAns);
}
public void actionPerformed(ActionEvent ae)
{
    String s="";
    boolean b;
    FileInputStream Fin;
    DataInputStream dis;
    FileOutputStream Fout;
    DataOutputStream dos;
    try
    {
        Fout = new FileOutputStream("Biodata.txt",true);
        dos = new DataOutputStream(Fout);
        String str = ae.getActionCommand();
        if(str.equals("SUBMIT"))
        {
            s=txtName.getText().trim();
            dos.writeUTF(s);
            dos.writeInt(Integer.parseInt(txtAge.getText()));
            s=txtAddr.getText();
            dos.writeUTF(s);
            if(chkMale.getState())
            dos.writeUTF("Male ");
            if(chkFemale.getState())
            dos.writeUTF("Female ");
            s="";
            if(chkMca.getState())
            s="MCA ";
            if(chkBca.getState())
            s+="BCA ";
            if(chkBba.getState())
            s+="BBA ";
            if(chkMba.getState())
            s+="MBA ";
            s+="!";
            dos.writeUTF(s);
            Fout.close();
        }
        if(str.equals("VIEW"))
        {
            String tmp,name,addr,gender,qual;
            int age;
            Fin = new FileInputStream("Biodata.txt");
```

```
dis = new DataInputStream(Fin);
inti=0,j;
while(Fin.available()>0)
{
    name = dis.readUTF();
    age =dis.readInt();
    addr = dis.readUTF();
    gender = dis.readUTF();
    qual = dis.readUTF();
    if(name.equals(txtName.getText().trim()))
    {
        txtAge.setText(age+"");
        txtAddr.setText(addr);
        if(gender.equals("Male "))
            chkMale.setState(true);
        else
            chkFemale.setState(true);
        while(qual.charAt(i)!=' ')
        {
            j=qual.indexOf(' ');
            tmp = qual.substring(i,j);
            if(tmp.equals("MCA"))
                chkMca.setState(true);
            if(tmp.equals("BCA"))
                chkBca.setState(true);
            if(tmp.equals("BBA"))
                chkBba.setState(true);
            if(tmp.equals("MBA"))
                chkMba.setState(true);
            i=j+1;
        }
        break;
    }
}
Fin.close();
}
if(str.equals("NEW"))
{
    txtName.setText("");
    txtAge.setText("");
    txtAddr.setText("");
    chkMale.setState(false);
    chkFemale.setState(false);
    chkMca.setState(false);
    chkBca.setState(false);
    chkBba.setState(false);
}
```

```

chkMba.setState(false);
    }
}
catch(Exception e)
{
System.out.println("The Exception Is : " +e);
}
}
}
class Bio2
{
public static void main(String args[])
{
try{
    Frame1 F = new Frame1("Biodata");
F.setSize(400,400);
F.show();
}catch(Exception e)
{
System.out.println(e);
}
}
}

```

(b) Design a calculator based on AWT application.

```

import java.awt.*;
import java.awt.event.*;
public class calculator implements ActionListener
{
    int c,n;
    String s1,s2,s3,s4,s5;
    Frame f;
    Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17;
    Panel p;
    TextField tf;
    GridLayout g;
    calculator()
    {
        f = new Frame("My calculator");
        p = new Panel();
        f.setLayout(new FlowLayout());
        b1 = new Button("0");
        b1.addActionListener(this);
        b2 = new Button("1");
        b2.addActionListener(this);
        b3 = new Button("2");
        b3.addActionListener(this);

```

```
b4 = new Button("3");
b4.addActionListener(this);
b5 = new Button("4");
b5.addActionListener(this);
b6 = new Button("5");
b6.addActionListener(this);
b7 = new Button("6");
b7.addActionListener(this);
b8 = new Button("7");
b8.addActionListener(this);
b9 = new Button("8");
b9.addActionListener(this);
b10 = new Button("9");
b10.addActionListener(this);
b11 = new Button("+");
b11.addActionListener(this);
b12 = new Button("-");
b12.addActionListener(this);
b13 = new Button("*");
b13.addActionListener(this);
b14 = new Button("/");
b14.addActionListener(this);
b15 = new Button("%");
b15.addActionListener(this);
b16 = new Button("=");
b16.addActionListener(this);
b17 = new Button("C");
b17.addActionListener(this);
tf = new TextField(20);
f.add(tf);
g = new GridLayout(4,4,10,20);
p.setLayout(g);
p.add(b1);p.add(b2);p.add(b3);p.add(b4);p.add(b5);p.add(b6);p.add(b7);p.add(b8);p.ad
d(b9);
p.add(b10);p.add(b11);p.add(b12);p.add(b13);p.add(b14);p.add(b15);p.add(b16);p.add(
b17);
f.add(p);
f.setSize(300,300);
f.setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==b1)
    {
        s3 = tf.getText();
        s4 = "0";
```



```
s5 = s3+s4;
tf.setText(s5);
}
if(e.getSource()==b2)
{
    s3 = tf.getText();
    s4 = "1";
    s5 = s3+s4;
    tf.setText(s5);
}
if(e.getSource()==b3)
{
    s3 = tf.getText();
    s4 = "2";
    s5 = s3+s4;
    tf.setText(s5);
}if(e.getSource()==b4)
{
    s3 = tf.getText();
    s4 = "3";
    s5 = s3+s4;
    tf.setText(s5);
}
if(e.getSource()==b5)
{
    s3 = tf.getText();
    s4 = "4";
    s5 = s3+s4;
    tf.setText(s5);
}
if(e.getSource()==b6)
{
    s3 = tf.getText();
    s4 = "5";
    s5 = s3+s4;
    tf.setText(s5);
}
if(e.getSource()==b7)
{
    s3 = tf.getText();
    s4 = "6";
    s5 = s3+s4;
    tf.setText(s5);
}
if(e.getSource()==b8)
{
```

```
s3 = tf.getText();
s4 = "7";
s5 = s3+s4;
tf.setText(s5);
}
if(e.getSource()==b9)
{
    s3 = tf.getText();
    s4 = "8";
    s5 = s3+s4;
    tf.setText(s5);
}
if(e.getSource()==b10)
{
    s3 = tf.getText();
    s4 = "9";
    s5 = s3+s4;
    tf.setText(s5);
}
if(e.getSource()==b11)
{
    s1 = tf.getText();
    tf.setText("");
    c=1;
}
if(e.getSource()==b12)
{
    s1 = tf.getText();
    tf.setText("");
    c=2;
}
if(e.getSource()==b13)
{
    s1 = tf.getText();
    tf.setText("");
    c=3;
}
if(e.getSource()==b14)
{
    s1 = tf.getText();
    tf.setText("");
    c=4;
}
if(e.getSource()==b15)
{
```

```
s1 = tf.getText();
tf.setText("");
c=5;
}
if(e.getSource()==b16)
{
    s2 = tf.getText();
    if(c==1)
    {
        n = Integer.parseInt(s1)+Integer.parseInt(s2);
        tf.setText(String.valueOf(n));
    }
    else
    if(c==2)
    {
        n = Integer.parseInt(s1)-Integer.parseInt(s2);
        tf.setText(String.valueOf(n));
    }
    else
    if(c==3)
    {
        n = Integer.parseInt(s1)*Integer.parseInt(s2);
        tf.setText(String.valueOf(n));
    }
    if(c==4)
    {
        try
        {
            int p=Integer.parseInt(s2);
            if(p!=0)
            {
                n = Integer.parseInt(s1)/Integer.parseInt(s2);
                tf.setText(String.valueOf(n));
            }
            else
                tf.setText("infinite");
        }
        catch(Exception i){}
    }
    if(c==5)
    {
        n = Integer.parseInt(s1)%Integer.parseInt(s2);
        tf.setText(String.valueOf(n));
    }
}
if(e.getSource()==b17)
```

```
        {
            tf.setText("");
        }
    }
    public static void main(String[] abc)
    {
        calculator v = new calculator();
    }
}
```

(c). Design an AWT application to generate result marks sheet.

```
import java.util.Scanner;
import java.lang.Exception;
class InvalidMarksException extends Exception
{
    InvalidMarksException(String message)
    {
        super(message);
    }
}
class variables
{
    int i, k;
    int marks[]=new int[10];
    int mark=0;
    float total=0;
}
interface sports
{
    int sports=10;
}
class student extends variables implements sports
{
    int detail()
    {
        Scanner inp=new Scanner(System.in);
        for(i=0;i<6;i++)
        {
            try
            {
                mark=inp.nextInt();
                if(mark>=0 && mark<=100)
                {
                    marks[i]=mark;
                    total=total+marks[i];
                }
            }
            else
```

```
        {
i=i-1;
throw new InvalidMarksException("INVALID MARKS INPUT ERROR");
        }
    }
    catch(InvalidMarksException e)
    {
        System.out.println(e.getMessage());
    }
}
for(i=0;i<6;i++)
{
    if(marks[i]<=50)
    {
        k=1;
    }
}
if(k==1)
{
    return 1;
}
else
{
    return 0;
}
}
int status()
{
    return k;
}
float grade()
{
    System.out.println("SPOATS MARKS OF 10 HAS BEEN ADDED TO THE TOTAL");
    total=total+10;
    floatavg=(total/600)*100;
    System.out.println("TOTAL is : "+total);
    System.out.println("PERCENTAGE is : "+avg);
    returnavg;
}
}
class Student1
{
    public static void main(String args[])
    {
        Scanner inp=new Scanner(System.in);
        student student1=new student();
    }
}
```

```
int usn, a, i, c;
float b;
String name;
System.out.println("Enter the NAME of the student");
name=inp.nextLine();
System.out.println("Enter the USN of the student");
usn=inp.nextInt();
for(;;)
{
System.out.println("ENTER \n 1: STUDENT MARKS 2: GENERATE REPORT 3: EXIT");
i=inp.nextInt();
switch(i)
{
case 1 :System.out.println("Enter the Marks of the student");
a=student1.detail();
break;
case 2:c=student1.status();
b=student1.grade();
if(c==1)
{
System.out.println("The Student : "+name+" with USN : "+usn+" Has FAILED");
}
else
{
System.out.println("The Student : "+name+" with USN : "+usn+" Has PASSED");
}
if((b>=80&&b<=100)&&c==0)
{
System.out.println("YOU ARE HONOURED FCD");
}
if((b>=70&&b<80)&&c==0)
{
System.out.println("YOU ARE HONOURED FC");
}
if((b>=60&&b<70)&&c==0)
{
System.out.println("YOU ARE HONOURED SC");
}
if((b>=50&&b<60)&&c==0)
{
System.out.println("YOU ARE HONOURED TC");
}
if((b>=40&&b<=100)&&c==1)
{
System.out.println("YOU ARE HONOURED FAIL");
}
```

```
break;  
case 3:  
System.out.println();  
System.out.println("THANK YOU");  
System.exit(0);  
default:  
System.out.println("TRY AGAIN");  
    }  
    }  
}  
}
```