1. Write a Java program to calculate gross salary net salary taking the following data.
   DA=40%of basic; HRA=20%of basic; CCA=Rs250/-;
   PF=10%of basic; PT=Rs100/- ; Income tax = 10% of gross;
   Gross income: Basic + DA + HRA + CCA;
   Deductions = PF+PT+IT;
   Net income = Gross income – Deductions.

Program

```
import java.util.Scanner;
public class EmployeeSalary
{
	public static void main (String args[])
	{
		String name,eid;
		double bsalary;
		Scanner sc = new Scanner (System.in);
		System.out.println ("Enter Name & Employee Id");
		name = sc.nextLine ();
		eid = sc.nextLine ();
		System.out.println ("Enter Basic Salary");
		bsalary = sc.nextDouble ();
		double da = (0.4)*bsalary;
		double hra = (0.2)*bsalary;
		double cca = 250;
		double pf = (0.1)*bsalary;
		double pt = 100;
		double gincome = bsalary+da+hra+cca;
		double it = (0.1)*gincome;
		double deduction = pf+pt+it;
		double nincome = gincome-deduction;
		System.out.println ("The salary of employee "+name+" with Id "+eid+" is
		Rs."+nincome);
	}
}
```

Output

```
Enter Name & Employee Id
ABC
7777
Enter Basic Salary
10000
The salary of employee ABC with Id 7777 is Rs.13525.0
```

2. Write a Java program that prints all real solutions to the quadratic equation ax2 + bx + c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions.

Program

```
import java.util.Scanner;
public class QuadraticEquation
{
	int a,b,c;
	double r1,r2,d;
	Scanner sc = new Scanner (System.in);

	void input ()
	{
		System.out.println ("Enter the values of a,b & c");
		a = sc.nextInt ();
		b = sc.nextInt ();
		c = sc.nextInt ();
	}

	void discriminant ()
	{
		d = (b*b)-4*a*c;
	}

	void calculateRoots ()
	{
		if(d > 0)
		{
			System.out.println ("Roots are Distant");
			r1 = (-b+Math.sqrt(d)) / (2*a);
			r2 = (-b-Math.sqrt(d)) / (2*a);
			System.out.println ("Roots are "+r1+" & "+r2);
		}
		else if(d == 0)
		{
			System.out.println ("Roots are Equal");
			r1 = (-b+Math.sqrt(d)) / (2*a); r2 = r1;
			System.out.println ("Roots are "+r1+" & "+r2);
		}
		else
		{
			System.out.print ("Roots are Imaginary\nNo Real Solutions");
		}
	}
}
```

```
class QERun
{
	public static void main (String args[])
	{
		QuadraticEquation qe = new QuadraticEquation ();
				qe.input ();
				qe.discriminant ();
				qe.calculateRoots ();
	}
}
```

Output

1. Enter the values of a,b & c
   2 6 1
   Roots are Distant
   Roots are -0.17712434446770464 & -2.8228756555322954

2. Enter the values of a,b & c
   2 4 2
   Roots are Equal
   Roots are -1.0 & -1.0

3. Enter the values of a,b & c
   7 4 2
   Roots are Imaginary
   No Real Solutions

3. Write a Java program to multiply two given matrices.

Program

```
import java.util.Scanner;
public class MatrixMultiplication
{
	int m,n,p,q,i,j,k=0;
	int a[][] = new int [10][10];
	int b[][] = new int [10][10];
	int c[][] = new int [10][10];
	Scanner sc = new Scanner (System.in);

	void Input ()
	{
		System.out.println ("Enter no. of Rows & Columns of 1st Matrix");
		m = sc.nextInt();
		n = sc.nextInt();
		System.out.println ("Enter no. of Rows & Columns of 2nd Matrix");
		p = sc.nextInt();
		q = sc.nextInt();
		if (n != p)
		{
			System.out.println("Matrix Multiplication is not possible");
			System.exit(0);
		}
		else
		{
			System.out.println("Enter elements of 1st Matrix");
			for (i=0;i<m;i++)
				for (j=0;j<n;j++)
					a[i][j] = sc.nextInt();
			System.out.println("Enter elements of 2nd Matrix");
			for (i=0;i<p;i++)
				for (j=0;j<q;j++)
					b[i][j] = sc.nextInt();
		}
	}

	void Calculate ()
	{
		for (i=0;i<m;i++)
			for (j=0;j<q;j++)
			{
				c[i][j] = 0;
				for (k=0;k<n;k++)
					c[i][j] = c[i][j] +a[i][k]*b[k][j];
			 }
	}
```

```
        void Result ()
        {
                System.out.println ("Product is");
                for (int i=0;i<m;i++)
                {
                        For ( int j=0;j<q;j++)
                        {
                                System.out.print (c[i][j]+"\t");
                        }
                        System.out.print ("\n");
                }
        }
}


class MulRun
{
        public static void main (String args [])
        {
                MatrixMultiplication mu = new MatrixMultiplication ();
                mu.Input();
                mu.Calculate();
                mu.Result();
        }
}
```

Output

1. Enter no. of Rows & Columns of 1st Matrix
   2 3
   Enter no. of Rows & Columns of 2nd Matrix
   3 2
   Enter elements of 1st Matrix
   4 9 7
   6 1 3
   Enter elements of 2nd Matrix
   2 7
   6 5
   1 8
   Product is
   71 136
   21 71

2. Enter no. of Rows & Columns of 1st Matrix
   2 4
   Enter no. of Rows & Columns of 2nd Matrix
   3 5
   Matrix Multiplication is not possible

4. Write a Java program to add and subtract two complex numbers and using the concept of constructor overloading.

Program

```
public class ComplexNumbers
{
	double r;
	double i;

	ComplexNumbers ()
	{
		r = 0;
		i = 0;
	}

	ComplexNumbers (double var )
	{
		r = var;
		i = 0;
	}

	ComplexNumbers (double var1,double var2)
	{
		r = var1;
		i = var2;
	}

	ComplexNumbers (ComplexNumbers obj)
	{
		r = obj.r;
		i = obj.i;
	}

	void add (ComplexNumbers obj1,ComplexNumbers obj2)
	{
		r = obj1.r+obj2.r;
		i = obj1.i+obj2.i;
		System.out.println ("Sum = "+r+"+"+i+"i");
	}

	void sub (ComplexNumbers obj1,ComplexNumbers obj2)
	{
		r = obj1.r-obj2.r;
		i = obj1.i-obj2.i;
		System.out.println ("Difference = "+r+"-"+i+"i");
	}
}
```

```
class CNRun
{
	public static void main (String args[])
	{
		ComplexNumbers c1 = new ComplexNumbers ();
		ComplexNumbers c2 = new ComplexNumbers (10);
		ComplexNumbers c3 = new ComplexNumbers (20,30);
		ComplexNumbers c4 = new ComplexNumbers (c1);
		c1.add (c1,c2);
		c2.sub (c3,c4);
	}
}
```

Output

```
Sum = 10.0+0.0i
Difference = 20.0-30.0i
```

5. Write a program to sort a list of elements in ascending and descending order and show exception handling.

Program

```
import java.util.Scanner;
public class Sort
{
	void sortAscend (int a[], int n)
	{
		for(int i=0;i<n-1;i++)
		    for(int j=0;j<n-1;j++)
		      if(a[j] > a[j+1])
		      {
		        int temp = a[j];
		            a[j] = a[j+1];
		            a[j+1] = temp;
		      }
	}

	void sortDescend (int a[], int n)
	{
		for(int i=0;i<n-1;i++)
		    for(int j=0;j<n-1;j++)
		      if(a[j] < a[j+1])
		      {
		        int temp = a[j];
		            a[j] = a[j+1];
		            a[j+1] = temp;
		      }
	}

	void printArray (int a[], int n)
	{
		for (int i=0;i<n;i++)
			 System.out.println (a[i]+" ");
	}
}

class SortRun
{
	public static void main (String args[])
	{
		int n;
		Scanner sc = new Scanner (System.in);
		Sort s = new Sort ();
		System.out.println ("Enter the no. of Elements");
		n = sc.nextInt();
		int array[] = new int [5];
```

```
            System.out.println ("Enter "+n+" Integers");
            for (int i=0;i<n;i++)
            {
                try
                {
                    array[i] = sc.nextInt();
                }
                catch (ArrayIndexOutOfBoundsException e)
                {
                    System.out.println ("Index out of Bounds "+e);
                    System.exit(0);
                }
            }
            System.out.println ("Entered Elements are");
            s.printArray(array, n);
            System.out.println ("Ascending Order:");
            s.sortAscend(array, n);
            s.printArray(array, n);
            System.out.println ("Descending Order:");
            s.sortDescend(array, n);
            s.printArray(array, n);
    }
}
```

Output

1. Enter the no. of Elements
   5
   Enter 5 Integers
   23 9 67 43 7
   Entered Elements are
   23
   9
   67
   43
   7
   Ascending Order:
   7
   9
   23
   43
   67
   Descending Order:
   67
   43
   23
   9
   7

2. Enter the no. of Elements
   7
   Enter 7 Integers
   23
   9
   32
   56
   78
   143
   Index out of Bounds
   java.lang.ArrayIndexOutOfBoundsException:
   Index 5 out of bounds for length 5

6. Write a program to calculate the interest amount based on the rate of interest defined for different banks using the concept of interface. Also calculate and display the maturity amount.

Program

```
import java.util.Scanner;
interface Bank
{
	Scanner sc = new Scanner (System.in);
	void getDetails ();
	void calculateInterest ();
	void getMAmount ();
}

class SBI implements Bank
{
	String name;
	double principal;
	double period;
	double interest;
	double roi = 8.5;

	public void getDetails ()
	{
		System.out.println ("Enter Name");
		name = sc.nextLine();
		System.out.println ("Enter Principal");
		principal = sc.nextDouble();
		System.out.println ("Enter Period");
		period = sc.nextDouble();
	}

	public void calculateInterest ()
	{
		interest = (principal * period * roi)/100;
	}

	public void getMAmount ()
	{
		double mAmount = principal + interest;
		System.out.println (name+" your Maturity Amount in SBI is Rs.
		"+mAmount+"\n");
	}
}
```

```
class CanaraBank implements Bank
{
	String name;
	double principal;
	double period;
	double interest;
	double roi = 8.5;

	public void getDetails ()
	{
		System.out.println ("Enter Name");
		name = sc.next();
		System.out.println ("Enter Principal");
		principal = sc.nextDouble();
		System.out.println ("Enter Period");
		period = sc.nextDouble();
	}

	public void calculateInterest ()
	{
		interest = (principal * period * roi)/100;
	}

	public void getMAmount ()
	{
		double mAmount = principal + interest;
		System.out.println (name+" your Maturity Amount in Canara Bank is Rs.
		"+mAmount+"\n");
	}
}

class BRun
{
	public static void main (String args[])
	{
		Bank b;
		SBI sb = new SBI ();
		CanaraBank cb = new CanaraBank ();
		b = sb;
		b.getDetails ();
		b.calculateInterest ();
		b.getMAmount ();
		b = cb;
		b.getDetails ();
		b.calculateInterest ();
		b.getMAmount ();
	}
}
```

Output

```
Enter Name
ABC
Enter Principal
25000
Enter Period
5
ABC your Maturity Amount in SBI is Rs. 35625.0

Enter Name
XYZ
Enter Principal
15000
Enter Period
10
XYZ your Maturity Amount in Canara Bank is Rs. 27750.0
```

7. Write a Java program to compute the surface area and volume of cylinder, cone and sphere. Create an abstract class “Solid” and the classes cylinder, cone and sphere have to inherit the common properties form the class “Solid”.

Program

```
import java.util.Scanner;
abstract class Solid
{
	double r;
	double h;
	abstract void SArea ();
	abstract void Vol ();

	void input ()
	{
		Scanner sc = new Scanner (System.in);
		System.out.println ("Enter the Radius");
		r = sc.nextDouble ();
		System.out.println ("Enter the Height");
		h = sc.nextDouble ();
	}
}

class Cylinder extends Solid
{
	void SArea ()
	{
		double sarea = (2*3.14*r*h) + (2*3.14*r*r);
		System.out.println ("Surface Area of Cylinder = "+sarea);
	}
	void Vol()
	{
		double vol = 3.14*r*r*h;
		System.out.println ("Volume of Cylinder = "+vol+"\n");
	}
}

class Cone extends Solid
{
	void SArea ()
	{
		double sarea = 3.14*r*(r+Math.sqrt(r*r+h*h));
		System.out.println ("Surface Area of Cone = "+sarea);
	}
```

```
	void Vol ()
	{
		double vol = (3.14*r*r*h) / 3;
		System.out.println ("Volume of Cone = "+vol+"\n");
	}
}

class Sphere extends Solid
{
	void SArea ()
	{
		double sarea = 4*3.14*r*r;
		System.out.println ("Surface Area of Sphere = "+sarea);
	}
	void Vol ()
	{
		double vol = (4/3)*3.14*r*r*r;
		System.out.println ("Volume of Sphere = "+vol+"\n");
	}
}

class SRun
{
	public static void main (String args[])
	{
		Solid s;
		Cylinder cy = new Cylinder ();
		s = cy;
		s.input ();
		s.SArea ();
		s.Vol ();
		Cone co = new Cone ();
		s = co;
		s.input ();
		s.SArea ();
		s.Vol ();
		Sphere sp = new Sphere ();
		s = sp;
		s.input ();
		s.SArea ();
		s.Vol ();
	}
}
```

Output

```
Enter the Radius
4
Enter the Height
7
Surface Area of Cylinder = 276.32
Volume of Cylinder = 351.68

Enter the Radius
5
Enter the Height
6
Surface Area of Cone = 201.1209199117345
Volume of Cone = 157.0

Enter the Radius
5
Enter the Height
5
Surface Area of Sphere = 314.0
Volume of Sphere = 392.5
```

8.

a. Write a Java program that creates three threads. First thread displays “Good Morning “every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.

Program

```
public class ThreadOne extends Thread
{
	public void run ()
	{
		try
		{
			for (int i=0;i<4;i++)
			{
				Thread.sleep(1000);
				System.out.println ("Good Morning");
			}
		}
		catch (InterruptedException e)
		{
			System.out.println ("Thread Interrupted");
		}
	}
}

public class ThreadTwo extends Thread
{
	public void run ()
	{
		try
		{
			for (int i=0;i<4;i++)
			{
				Thread.sleep(2000);
				System.out.println ("Hello");
			}
		}
		catch (InterruptedException e)
		{
			System.out.println ("Thread Interrupted");
		}
	}
}
```

```
public class ThreadThree extends Thread
{
	public void run ()
	{
		try
		{
			for (int i=0;i<4;i++)
			{
				Thread.sleep(3000);
				System.out.println ("Welcome");
			}
		}
		catch (InterruptedException e)
		{
			System.out.println ("Thread Interrupted");
		}
	}
}

public class ProgramMultiThread
{
	public static void main (String args[])
	{
		ThreadOne t1 = new ThreadOne ();
		ThreadTwo t2 = new ThreadTwo ();
		ThreadThree t3 = new ThreadThree ();

		t1.start ();
		t2.start ();
		t3.start ();

		try
		{
			//Thread.sleep (15000);
			t1.join ();
			t2.join ();
			t3.join ();
		}
		catch (InterruptedException e)
		{
			System.out.println ("Thread Interrupted");
		}
		System.out.println ("Main Thread Exits");
	}
}
```

Output

Good Morning
Hello
Good Morning
Welcome
Good Morning
Hello
Good Morning
Hello
Welcome
Hello
Welcome
Welcome
Main Thread Exits

b. Write a Java program to read from console and write these data into file, again read that file and print the content onto console.

Program

```
import java.io.*;

public class IOExample
{
	public static void main (String args[]) throws IOException,FileNotFoundException
	{
		char c;
		int i;
		FileInputStream fin = new FileInputStream ("/Users/gautamkumar/eclipse-
workspace/Lab Programs/src/Test.txt"); //Set path of Test.txt
		FileOutputStream fout = new FileOutputStream
("/Users/gautamkumar/eclipse-workspace/Lab Programs/src/Test.txt"); //Set path of Test.txt
		BufferedReader br = new BufferedReader (new InputStreamReader
(System.in));
		System.out.println ("Enter the characters: q to quit");

		do {
			c = (char)br.read();
			if (c != 'q')
				fout.write(c);
		} while (c != 'q');

		do {
			i = fin.read();
			if (i != -1)
				System.out.print ((char)i);
		} while (i != -1);
		fin.close();
		fout.close();
	}
}
```

Output

```
Enter the characters: q to quit
HelloWorld q
HelloWorld
```

9. Write a java program to implement a stack using generic class and methods.

Program

```
import java.util.Scanner;
public class Stack <E>
{
	E stck[];
	int top;
	final int size = 10;
	@SuppressWarnings("unchecked")

	Stack()
	{
		stck = (E[]) new Object[size];
		top = -1;
	}

	void push (E item)
	{
		if (top == size-1)
			System.out.println ("Stack is Full");
		else
			stck[++top] = item;
	}

	E pop ()
	{
		if (top < 0)
		{
			System.out.println ("Stack Underflow");
			return null;
		}
		else
			return stck[top--];
	}
}

class TestStack
{
	public static void main (String args[])
	{
		Stack <Integer> si = new Stack <Integer> ();
		Stack <Double> sd = new Stack <Double> ();

		Scanner sc = new Scanner (System.in);
		System.out.println ("Enter the elements of Integer Stack");
		for (int i=0;i<10;i++)
```

```
		{
			int n = sc.nextInt ();
			si.push(n);
		}
		System.out.println ("Enter the elements of Double Stack");
		for (int i=0;i<10;i++)
		{
			Double n = sc.nextDouble ();
			sd.push(n);
		}

		System.out.println ("Elements of Integer Stack are:");
		for (int i=0;i<10;i++)
			System.out.println (si.pop());
		System.out.println ("Elements of Double Stack are:");
		for (int i=0;i<10;i++)
			System.out.println (sd.pop());
	}
}
```

Output

```
Enter the elements of Integer Stack
2 18 5 25 69 43 23 5 34 7
Enter the elements of Double Stack
2.5 32.1 56.7 69.54 43.6 65.45 32.65 2.5 15.6 4.6
Elements of Integer Stack are:
7
34
5
23
43
69
25
5
18
2
Elements of Double Stack are:
4.6
15.6
2.5
32.65
65.45
43.6
69.54
56.7
32.1
2.5
```

10. Write a Java program to read a string from the keyboard and to do the following

a. Extract the middle character of the string

Program

```
import java.util.Scanner;
public class MidString
{
	public static void main (String args[])
	{
		String s;
		int len,mid;
		Scanner sc = new Scanner (System.in);
		System.out.println ("Enter a String");
		s = sc.nextLine();
		len = s.length();
		mid = len/2;
		if (len%2 != 0)
			System.out.println ("Middle Character is "+s.charAt(mid));
		else
			System.out.println ("Middle Character are "+s.charAt(mid-1)+" &
			"+s.charAt(mid));
	}
}
```

Output

1. Enter a String
   Hello
   Middle Character is l

2. Enter a String
   Computer
   Middle Character are p & u

b. Check whether the string entered is palindrome or not

Program

```
import java.util.Scanner;
public class Palindrome
{
	public static void main (String args[])
	{
		String original,rev = "";
		int i;
		Scanner sc = new Scanner (System.in);
		System.out.println ("Enter a String");
		original = sc.nextLine();
```

```
        for (i = original.length()-1;i>=0;i--)
        {
                rev = rev + original.charAt(i);
        }
        System.out.println ("Original String: "+original);
        System.out.println ("Reversed String: "+rev);
        if (original.equals(rev))
                System.out.println ("String is a Palindrome");
        else
                System.out.println ("String is not a Palindrome");
    }
}
```

Output

1. Enter a String
   Refer
   Original String: refer
   Reversed String: refer
   String is a Palindrome

2. Enter a String
   Hello
   Original String: Hello
   Reversed String: olleH
   String is not a Palindrome

c. Counting the number of vowels in the string

Program

```
import java.util.Scanner;
public class CountVowels
{
        public static void main (String args[])
        {
                String s;
                char ch;
                int count = 0;
                Scanner sc = new Scanner (System.in);
                System.out.println ("Enter a String");
                s = sc.nextLine();
                for (int i=0;i<s.length();i++)
                {
                        ch = s.charAt(i);
                        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
                                count++;
                }
                System.out.println ("No. of Vowels = "+count);
        }
}
```

Output

```
Enter a String
Computer Science
No. of Vowels = 6
```

d. Counting the total number of characters in this string.

Program

```
import java.util.Scanner;
public class CharacterCount
{
	public static void main (String args[])
	{
		String s;
		int count = 0;
		Scanner sc = new Scanner (System.in);
		System.out.println ("Enter a String");
		s = sc.nextLine();
		for (int i=0;i<s.length();i++)
		{
			if (s.charAt(i) != ' ')
				count++;
		}
		System.out.println ("No. of Alphanumeric Characters = "+count);
	}
}
```

Output

```
Enter a String
Object
No. of Alphanumeric Characters = 6
```