SIR M VISVESVARAYA INSTITUTE OF TECHNOLOGY

(Affiliated to VTU, approved by AICTE and Accredited by NBA, NAAC),Bengaluru – 562157

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

LABORATORY MANUAL

OBJECT ORIENTED PROGRAMMING WITH JAVA LABORATORY/21CSL35

(Effective from the academic year 2021 -2022)

Prepared by:-

Mr. Vijay kumar Y M, Assistant Professor, Dept. of ISE
Mr. Vitesh Babu M, Assistant Professor, Dept. of ISE

Sir M VISVESVARAYA INSTITUTE OF TECHNOLOGY

(Accredited by NAAC, Approved by A.I.C.T.E. New Delhi, Recognized by Govt. of Karnataka & Affiliated to V.T U., Belagavi)

Sri Krishnadevaraya Nagara, Hunasamaranahalli, Bengaluru- 562157

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

Institute Vision

- To be a centre of excellence in technical and management education concurrently focusing on disciplined and integrated development of personality through quality education, sports, cultural and co-curricular activities.
- To promote transformation of students into better human beings, responsible citizens and competent professionals to serve as a valuable resource for industry, work environment and society.

Institute Mission

- To impart quality technical education, provide state-of-art facilities, achieve high quality in teaching-learning & research and encourage extra & co-curricular activities.
- To stimulate in students a spirit of inquiry and desire to gain knowledge and skills to meet the changing needs that can enrich their lives.
- To provide opportunity and resources for developing skills for employability and entrepreneurship, nurturing leadership qualities, imbibing professional ethics and societal commitment.
- To create an ambiance and nurture conducive environment for dedicated and quality staff to upgrade their knowledge & skills and disseminate the same to students on a sustainable long term basis.
- To facilitate effective interaction with the industries, alumni and research institutions.

Department Vision

- To empower students with knowledge and skills to develop the competency in the emerging areas of Information Technology

Department Mission

- To train the students to have Professional career in IT industry and Higher studies ThroughQuality Education.
- To provide outstanding Teaching and Research environment by implementing innovative Teaching and Research Methodologies for Quality Education and Research.

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

	Program Outcomes
a.	**Engineering Knowledge:** Apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
b.	**Problem Analysis: Identify,** formulate, research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences
c.	**Design/ Development of Solutions**: Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societaland environmental considerations.
d.	**Conduct investigations of complex problems** using research-based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of information to provide valid conclusions.
e.	**Modern Tool Usage**: Create, select and apply appropriate techniques, resourcesand modern engineering and IT tools including prediction and modeling to Complex engineering activities with an under- standing of the limitations.
f.	**The Engineer and Society**: Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and theConsequent responsibilities relevant to professional engineering practice.
g.	**Environment and Sustainability**: Understand the impact of professional Engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.
h.	**Ethics**: Apply ethical principles and commit to professional ethics andResponsibilities and norms of engineering practice.
i.	**Individual and Team Work:** Function effectively as an individual, and as amember or leader in diverse teams and in multi disciplinary settings.
j.	**Communication:** Communicate effectively on complex engineering activities withthe engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions**.**
k.	**Life-long Learning:** Recognize the need for and have the preparation and ability to engage in independent and life- long learning in the broadest context of technological change.
l.	**Project Management and Finance:** Demonstrate knowledge and understandingof engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in Multi disciplinary environments.
Program Specific Outcomes	
m.	**PSO1**: Adapt, Contribute Innovate ideas in the field of Artificial Intelligence andMachine Learning
n.	**PSO2**: Enrich the abilities to qualify for Employment, Higher studies and Research in various domains of Artificial Intelligence and Machine Learning such as Data Science, Computer Vision, Natural Language Processing with ethical values
o.	**PSO3**: Acquire practical proficiency with niche technologies and open source platforms and become Entrepreneur in the domain of Artificial Intelligence and Machine Learning

List of Experiments

OBJECT ORIENTED PROGRAMMING WITH JAVA LABORATORY			
Course Code	**21CSL35**	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	0:0:2:0	SEE Marks	50
Total Hours of Pedagogy	24	Total Marks	100
Credits	1	Exam Hours	03

Course Objectives:

CLO 1. Demonstrate the use of Eclipse/Netbeans IDE to create Java Applications.
CLO 2. Using java programming to develop programs for solving real-world problems.CLO 3.Reinforce the understanding of basic object-oriented programming concepts.

	Note: two hours tutorial is suggested for each laboratory sessions.
	Prerequisite
	• Students should be familiarized about java installation and setting the java environment. • Usage of IDEs like Eclipse/Netbeans should be introduced.
Sl. No.	***PART A – List of problems for which student should develop program and execute in the Laboratory***
1	Aim: Introduce the java fundamentals, data types, operators in java Program: 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.
2	Aim: Demonstrating creation of java classes, objects, constructors, declaration andinitialization ofvariables. Program: Create a Java class called **Student** with the following details as variables within it.USNName BranchPhone Write a Java program to create n Student objects and print the USN, Name, Branch, and Phoneofthese objects with suitable headings.
3	Aim: Discuss the various Decision-making statements, loop constructs in java Program: A. Write a program to check prime number B. Write a program for Arithmetic calculator using switch case menu
4	Aim: Demonstrate the core object-oriented concept of Inheritance, polymorphism Design a super class called **Staff** with details as StaffId, Name, Phone, Salary. Extend this class bywriting three subclasses namely Teaching (domain, publications), Technical (skills), and Contract (period). Write a Java program to read and display at least 3 staff objects of all three categories.
5	Aim: Introduce concepts of method overloading, constructor overloading, overriding. Program: Write a java program demonstrating Method overloading and Constructoroverloading.
6	Aim: Introduce the concept of Abstraction, packages. Program: Develop a java application to implement currency converter (Dollar to INR, EURO to INR, Yen to INR and vice versa), distance converter (meter to KM, miles to KM and vice versa),time converter (hours to minutes, seconds and vice versa) using packages.

7	Aim: Introduction to abstract classes, abstract methods, and Interface in java Program: Write a program to generate the resume. Create 2 Java classes Teacher (data: personal information, qualification, experience, achievements) and Student (data: personal information, result, discipline) which implements the java interface Resume with the method biodata().
8	Aim: Demonstrate creation of threads using Thread class and Runnable interface, multi- threaded programming. Program: Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer for every 1 second; second thread computes thesquare of the number and prints; third thread will print the value of cube of the number.
9	Aim: Introduce java Collections. Program: Write a program to perform string operations using ArrayList. Write functions for the following a. Append - add at end b. Insert – add at particular index c. Search d. List all string starts with given letter.
10	Aim: Exception handling in java, introduction to throwable class, throw, throws, finally. Program: Write a Java program to read two integers a and b. Compute a/b and print, when b is notzero. Raise an exception when b is equal to zero.
11	Aim: Introduce File operations in java. Program: Write a java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of thefile in bytes
12	Aim: Introduce java Applet, awt, swings. Programs: Develop an applet that displays a simple message in center of the screen. Develop a simple calculatorusing Swings.
	PART B – Practical Based Learning
01	A problem statement for each batch is to be generated in consultation with the co-examiner and student should develop an algorithm, program and execute the program for the given problem with appropriate outputs.

Course Outcome (Course Skill Set)

At the end of the course the student will be able to:

CO 1. Use Eclipse/NetBeans IDE to design, develop, debug Java Projects.
CO 2. Analyze the necessity for Object Oriented Programming paradigm over structured programmingand become familiar with the fundamental concepts in OOP.
CO 3. Demonstrate the ability to design and develop java programs, analyze, and interpret object-orienteddata and document results.
CO 4. Apply the concepts of multiprogramming, exception/event handling, abstraction to developrobust programs.
CO 5. Develop user friendly applications using File I/O and GUI concepts.

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks). A student shallbe deemed to have satisfied the academic requirements and earned the credits allotted to each course. The student has to secure not less than 35% (18 Marks out of 50) in the semester-end examination (SEE).

Continuous Internal Evaluation (CIE):

CIE marks for the practical course is **50 Marks**.

The split-up of CIE marks for record/ journal and test are in the ratio **60:40**.

- Each experiment to be evaluated for conduction with observation sheet and record write-up.

Rubrics for the evaluation of the journal/write-up for hardware/software experiments designed by the faculty who is handling the laboratory session and is made known to students at the beginning of the practical session.

- Record should contain all the specified experiments in the syllabus and each experiment write-up willbe evaluated for 10 marks.
- Total marks scored by the students are scaled downed to 30 marks (60% of maximum marks).
- Weightage to be given for neatness and submission of record/write-up on time.
- Department shall conduct 02 tests for 100 marks, the first test shall be conducted after the 8th weekof thesemester and the second test shall be conducted after the 14th week of the semester.
- In each test, test write-up, conduction of experiment, acceptable result, and procedural knowledge willcarry a weightage of 60% and the rest 40% for viva-voce.
- The suitable rubrics can be designed to evaluate each student's performance and learning ability.Rubrics suggested in Annexure-II of Regulation book
- The average of 02 tests is scaled down to **20 marks** (40% of the maximum marks).

The Sum of scaled-down marks scored in the report write-up/journal and average marks of two tests is the totalCIE marks scored by the student.

Semester End Evaluation (SEE):

- SEE marks for the practical course is 50 Marks.
- SEE shall be conducted jointly by the two examiners of the same institute, examiners areappointed by the University
- All laboratory experiments are to be included for practical examination.
- (Rubrics) Breakup of marks and the instructions printed on the cover page of the answer scriptto be strictly adhered to by the examiners. **OR** based on the course requirement evaluation rubrics shall be decided jointly by examiners.
- Students can pick one question (experiment) from the questions lot prepared by the internal /external examiners jointly.
- Evaluation of test write-up/ conduction procedure and result/viva will be conducted jointly by examiners.
- General rubrics suggested for SEE are mentioned here, writeup-20%, Conduction procedure and resultin -60%, Viva-voce 20% of maximum marks. SEE for practical shall be evaluated for 100 marks and scored marks shall be scaled down to 50 marks (however, based on course type, rubrics shall be decided by the examiners)
- *Students can pick one experiment from the questions lot of PART A with equal choice to all the studentsin a batch. For PART B examiners should frame a question for each batch, student should develop an algorithm, program, execute and demonstrate the results with appropriate output for the given problem.*
- *Weightage of marks for PART A is 80% and for PART B is 20%. General rubrics suggested to be followed for part A and part B.*
- Change of experiment is allowed only once and Marks allotted to the procedure part to bemadezero (Not allowed for Part B).
- The duration of SEE is 03 hours
- Rubrics suggested in Annexure-II of Regulation book

Suggested Learning Resources:

1. E Balagurusamy, Programming with Java, Graw Hill, 6th Edition, 2019.
2. Herbert Schildt, C: Java the Complete Reference, McGraw Hill, 11th Edition, 2020

PROGRAM :1

1. 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.

Aim: Introduce the java fundamentals, data types, operators in java

PROGRAM :

```
import java.util.Scanner;
public class Quadratic
{
        public static void main(String[] args)
        {

            int a, b, c;                                        // coefficients
            double root1, root2;
            System.out.println("Enter the coefficients");
            Scanner in=new Scanner(System.in);
           a = in.nextInt();
           b = in.nextInt();
           c = in.nextInt();

                                                                // calculate the determinant (b2 - 4ac)
           double d = b * b - 4 * a * c;
           System.out.println("Determinant="+d);
            if (d > 0)                                          // check if determinant is greater than 0
           {

                                                                // two real and distinct roots
           root1 = (-b + Math.sqrt(d)) / (2 * a);
           root2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.println("The roots are real and distinct");
            System.out.format("root1 = %.2f and root2 = %.2f", root1, root2);
           }

          else if (d == 0)                                      // check if determinant is equal to 0
          {

            // two real and equal roots
             // determinant is equal to 0
             // so -b + 0 == -b
           root1 = root2 = -b / (2 * a);
           System.out.println("The roots are real and equal");
           System.out.format("root1 = root2 = %.2f;", root1);
          }

          else                                                  // if determinant is less than zero
         {

            // roots are complex number and distinct
```

```
                double real = -b / (2 * a);
                double imaginary = Math.sqrt(-d) / (2 * a);
                System.out.println("The roots are imaginary");
                System.out.format("root1 = %.2f+%.2fi", real, imaginary);
                System.out.format("\nroot2 = %.2f-%.2fi", real, imaginary);
            }
        }
}
```

OUT PUT:

```
C:\javasample\JAVALAB>javac Quadratic.java

C:\javasample\JAVALAB>java Quadratic
Enter the coefficients
1
-5
2
Determinant=17.0
The roots are real and distinct
root1 = 4.56 and root2 = 0.44
```

```
C:\javasample\JAVALAB>javac Quadratic.java

C:\javasample\JAVALAB>java Quadratic
Enter the coefficients
3
2
1
Determinant=-8.0
The roots are imaginary
root1 = 0.00+0.47i
root2 = 0.00-0.47i
```

```
C:\javasample\JAVALAB>javac Quadratic.java

C:\javasample\JAVALAB>java Quadratic
Enter the coefficients
1
2
1
Determinant=0.0
The roots are real and equal
root1 = root2 = -1.00;
```

PROGRAM :2

2. Create a Java class called **Student** with the following details as variables within it.
USN
Name
Br nch
Phone
Write a Java program to create n Student objects and print the USN, Name, Branch, and Phoneof these objects with suitable headings.

Aim: Demonstrating creation of java classes, objects, constructors, declaration andinitialization of variables.

PROGRAM :

```
import java.util.*;
public class Student
{
    String usn,name,branch;
    long phone;

    void insertStudent(String reg,String nm, String br,long ph)
    {
            usn=reg;
           name=nm;
           branch=br;
           phone=ph;
     }

    void displayStudent()
    {
            System.out.println("**********************");
           System.out.println("USN= "+usn);
           System.out.println("NAME= "+name);
           System.out.println("BRANCH= "+branch);
           System.out.println("PHONE NUMBER= "+phone);
           System.out.println("**********************");
    }

    public static void main(String args[])
    {
           Student st[]=new Student [100];
           Scanner ip=new Scanner(System.in);
           System.out.println("Enter the number of students");
           int n=ip.nextInt();

           for(int i=0;i<n;i++)
                  st[i]=new Student();
           for(int j=0;j<n;j++)
           {
```

```
                    System.out.println("Enter the Usn,Name,Branch,Phone Number");
                     String usn=ip.next();
                    String name=ip.next();
                    String branch=ip.next();
                    long phone=ip.nextLong();
                     st[j].insertStudent(usn,name,branch,phone);

            }
            for( int m=0;m<n;m++)
            {
                     System.out.format("Student %d details are\n",m+1);
                     st[m].displayStudent();
            }

        }
}
```

OUTPUT:

```
C:\javasample\JAVALAB>javac Student.java

C:\javasample\JAVALAB>java Student
Enter the number of students
2
Enter the Usn,Name,Branch,Phone Number
1KT21CS001
Imran
CSE
9886663277
Enter the Usn,Name,Branch,Phone Number
1KT21AI002
Ramya
AIML
7795370211
Student 1 details are
************************
USN= 1KT21CS001
NAME= Imran
BRANCH= CSE
PHONE NUMBER= 9886663277
************************
Student 2 details are
************************
USN= 1KT21AI002
NAME= Ramya
BRANCH= AIML
PHONE NUMBER= 7795370211
************************
```

PROGRAM :3

3. A. Write a program to check prime number

Aim: Discuss the various Decision-making statements, loop constructs in java

PROGRAM :3A

```
import java.util.Scanner;
class Prime
{
    public static void main(String args[])
    {
    int i,n,flag=0;
    System.out.println("Enter the number");
    Scanner inp=new Scanner(System.in);
    n=inp.nextInt();
    for(i=2;i<n;i++)
    {
            if(n%i==0)
            {
                    flag=1;
                    break;
            }
    }
    if(flag==1)
            System.out.println("The given number is Not a Prime");
    else
            System.out.println("The given number is Prime");
    }
}
```

OUTPUT:

```
C:\javasample\JAVALAB>javac Prime.java

C:\javasample\JAVALAB>java Prime
Enter the number
8
The given number is Not a Prime

C:\javasample\JAVALAB>java Prime
Enter the number
9
The given number is Not a Prime

C:\javasample\JAVALAB>java Prime
Enter the number
7
The given number is Prime
```

PROGRAM :3B

3. B.Write a program for Arithmetic calculator using switch case menu

```
import java.util.*;
class Switch
{
      public static void main(String[] args)
      {
            Scanner inp = new Scanner(System.in);
            System.out.println("Enter the Operator (+,-,*,/) : ");
            char operator = inp.next().charAt(0);
            System.out.println("Enter the First Operand : ");
            double first = inp.nextDouble();
            System.out.println("Enter the Second Operand : ");
            double second = inp.nextDouble();
            double result = 0;
           switch(operator)
          {
            case '+':
               result = first + second;
               System.out.println("The Result is : "+first+" "+operator+" "+second+" = "+result);
               break;
            case '-':
               result = first - second;
               System.out.println("The Result is : \n "+first+" "+operator+" "+second+" = "+result);
               break;
            case '*':
               result = first * second;
               System.out.println("The Result is : "+first+" "+operator+" "+second+" = "+result);
               break;
            case '/':
               result = first / second;
               System.out.println("The Result is : \n "+first+" "+operator+" "+second+" = "+result);
               break;
            default :
               System.out.println("Invalid Operator");
               break;
            }
       }
}
```

OUTPUT:

```
C:\javasample\JAVALAB>javac Switch.java

C:\javasample\JAVALAB>java Switch
Enter the Operator (+,-,*,/) :
+
Enter the First Operand :
4
Enter the Second Operand :
4
The Result is : 4.0 + 4.0 = 8.0

C:\javasample\JAVALAB>java Switch
Enter the Operator (+,-,*,/) :
=
Enter the First Operand :
1
Enter the Second Operand :
1
Invalid Operator
```

PROGRAM :4

4. Design a super class called **Staff** with details as StaffId, Name, Phone, Salary. Extend this class by writing three subclasses namely Teaching (domain, publications), Technical (skills), and Contract (period). Write a Java program to read and display at least 3 staff objects of all three categories.

Aim: Demonstrate the core object-oriented concept of Inheritance, polymorphism

PROGRAM:

```
import java.util.Scanner;
class Staff
{
   String staffId;
   String name;
   long phone;
   float salary;
   public void accept()
   {
         Scanner scanner = new Scanner(System.in);
         System.out.print("Enter Staff Id: ");
         staffId = scanner.next();
         System.out.print("Enter Name: ");
         name = scanner.next();
         System.out.print("Enter Phone: ");
         phone = scanner.nextLong();
         System.out.print("Enter Salary: ");
         salary = scanner.nextFloat();

     }
      public void display()
      {
           System.out.println("Staff Id: " + staffId);
           System.out.println("Name: " + name);
           System.out.println("Phone: " + phone);
           System.out.println("Salary: " + salary);
      }
}
class Teaching extends Staff
{
   String domain;
   int n;
   public void accept()
   {
        super.accept();
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Domain:  ");
        domain = scanner.next();
        System.out.print("Enter Number of Publications: ");
```

```
            n = scanner.nextInt();
            System.out.println("\n");
    }
    public void display()
    {
          super.display();
          System.out.println("Domain: " + domain);
          System.out.println("Publications:"+n);
          System.out.println("\n");
    }
}
class Technical extends Staff
{
       String skill;
       public void accept()
       {
            super.accept();
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter technical Skills: ");
            skill = scanner.nextLine();
            System.out.println("\n");
       }
       public void display()
       {
            super.display();
            System.out.println("Technical Skills: " + skill);
            System.out.println("\n");
       }

}
class Contract extends Staff
{
    int period;
    public void accept()
    {
            super.accept();
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter Period: ");
            period = scanner.nextInt();
            System.out.println("\n");
    }
    public void display()
    {
            super.display();
            System.out.println("Contract Period: " + period);
    }
}
class Four
{
    public static void main(String[] args)
    {
```

```
        Teaching teaching = new Teaching();
        System.out.println("Enter the details of Teaching Staff");
        teaching.accept();

        Technical technical = new Technical();
        System.out.println("Enter the details of Technical Staff");
        technical.accept();

        Contract contract = new Contract();
        System.out.println("Enter the details of Contract Staff");
        contract.accept();

        System.out.println("The details of Teaching Staff");
        teaching.display();
        System.out.println("The details of Technical Staff");
        technical.display();
        System.out.println("The details of Contract Staff");
        contract.display();
    }
}
```

OUTPUT :

```
C:\javasample\JAVALAB>javac Four.java

C:\javasample\JAVALAB>java Four
Enter the details of Teaching Staff
Enter Staff Id: T100
Enter Name: Imran
Enter Phone: 9886663277
Enter Salary: 20000
Enter Domain: IOT
Enter Number of Publications: 5


Enter the details of Technical Staff
Enter Staff Id: TE100
Enter Name: Virat
Enter Phone: 9242125732
Enter Salary: 25000
Enter technical Skills: C,C++


Enter the details of Contract Staff
Enter Staff Id: C100
Enter Name: David
Enter Phone: 9242578612
Enter Salary: 22000
Enter Period: 2


The details of Teaching Staff
Staff Id: T100
Name: Imran
Phone: 9886663277
Salary: 20000.0
Domain: IOT
Publications:5


The details of Technical Staff
Staff Id: TE100
Name: Virat
Phone: 9242125732
Salary: 25000.0
Technical Skills: C,C++


The details of Contract Staff
Staff Id: C100
Name: David
Phone: 9242578612
Salary: 22000.0
Contract Period: 2
```

PROGRAM :5

5. Write a java program demonstrating Method overloading and Constructor overloading.

Aim: Introduce concepts of method overloading, constructor overloading.

PROGRAM:5A

Demonstrating Method overloading

```
class MOverloading
{
      //adding two integer numbers
      int add(int a, int b)
      {
                int sum = a+b;
                return sum;
      }
      //adding three integer numbers
       int add(int a, int b, int c)
      {
                int sum = a+b+c;
                return sum;
      }
      float add(float a, float b)
       {
                 float sum = a+b;
                 return sum;
       }
}
 class MOverload
{
       public static void main(String args[])
       {
                MOverloading obj = new MOverloading();
                int s1=obj.add(10, 20);
                int s2=obj.add(10, 20, 30);
                float s3=obj.add(2.2f,2.2f);
                System.out.println("Method Overload Sum1="+s1);
                System.out.println("Method Overload Sum2="+s2);
                System.out.println("Method Overload Sum3="+s3);
       }
}
```

OUTPUT:

```
C:\javasample\JAVALAB>javac MOverload.java

C:\javasample\JAVALAB>java MOverload
Method Overload Sum1=30
Method Overload Sum2=60
Method Overload Sum3=4.4
```

PROGRAM:5B

Constructor Overloading

```
public class Constructor
{
    int id;
    String name;
    Constructor()
    {
        System.out.println("This is Default constructor");
        System.out.println("Student Id : "+id + "\nStudent Name : "+name);
    }
    Constructor(int i, String n)
    {
         System.out.println("This is Parameterized Constructor:");
        id = i;
        name = n;
        System.out.println("Student Id : "+id + "\nStudent Name : "+name);
    }
    public static void main(String[] args)
    {
        Constructor s = new Constructor();
        Constructor student = new Constructor(10, "David");
    }
}
```

OUTPUT:

```
C:\javasample\JAVALAB>javac Constructor.java

C:\javasample\JAVALAB>java Constructor
This is Default constructor
Student Id : 0
Student Name : null
This is Parameterized Constructor:
Student Id : 10
Student Name : Imran
```

PROGRAM :6

6. Develop a java application to implement currency converter (Dollar to INR, EURO to INR, Yen to INR and vice versa), distance converter (meter to KM, miles to KM and vice versa), time converter (hours to minutes, seconds and vice versa) using packages.

Aim: Introduce the concept of Abstraction, packages.

PROGRAM:

CurrencyC.java

```
package cc;
import java.util.*;
public class CurrencyC
{
    double inr,usd;
    double euro,yen;
    Scanner in=new Scanner(System.in);
    public void dollartorupee()
    {
System.out.println("Enter dollars to convert into Rupees:");
usd=in.nextInt();
inr=usd*81.83;
System.out.println("Dollar ="+usd+" equal to INR="+inr);
System.out.println("\n");
    }
    public void rupeetodollar()
    {
        System.out.println("Enter Rupee to convert into Dollars:");
        inr=in.nextInt();
        usd=inr/81.83;
        System.out.println("Rupee ="+inr+"equal to Dollars="+usd);
    }
    public void eurotorupee()
    {
        System.out.println("Enter Euro to convert into Rupees:");
        euro=in.nextInt();
        inr=euro*79.06;
        System.out.println("Euro ="+euro+" equal to INR="+inr);
        System.out.println("\n");
    }
    public void rupeetoeuro()
    {
        System.out.println("Enter Rupees to convert into Euro:");
        inr=in.nextInt();
        euro=(inr/79.06);
        System.out.println("Rupee ="+inr +"equal to Euro="+euro);
        System.out.println("\n");
    }
```

```
    public void yentoruppe()
    {
        System.out.println("Enter Yen to convert into Rupees:");
        yen=in.nextInt();
        inr=yen*0.57;
        System.out.println("Yen ="+yen+" equal to INR="+inr);
        System.out.println("\n");
    }
    public void ruppetoyen()
    {
        System.out.println("Enter Rupees to convert into Yen:");
        inr=in.nextInt();
        yen=(inr/0.57);
        System.out.println("INR="+inr +"equal to YEN"+yen);
        System.out.println("\n");
    }
}
```

DistaceC.Java

```
package dc;
import java.util.*;
public class DistanceC
{
    double km,m,miles;
    Scanner in=new Scanner(System.in);
    public void mtokm()
    {
System.out.println("Enter the distance in meter");
m=in.nextDouble();
km=(m/1000);
System.out.println(m+"m" +" is equal to "+km+"km");
System.out.println("\n");
    }
    public void kmtom()
    {
        System.out.println("Enter the distance in Kilometer");
        km=in.nextDouble();
        m=km*1000;
        System.out.println(km+"km" +" is equal to "+m+"m");
        System.out.println("\n");
    }
    public void milestokm()
    {
        System.out.println("Enter the distance in miles");
        miles=in.nextDouble();
        km=(miles*1.60934);
        System.out.println(miles+"miles" +" is equal to "+km+"km");
        System.out.println("\n");
    }
```

```
    public void kmtomiles()
    {
            System.out.println("Enter the distance in km");
            km=in.nextDouble();
            miles=(km*0.621371);
            System.out.println(km+"km" +" is equal to "+miles+"miles");
    }
}
```

TimeC.java

```
package tc;
import java.util.*;
public class TimeC
{
    int hours,seconds,minutes;
    Scanner in = new Scanner(System.in);
    public void hourstominutes()
    {
            System.out.println("Enter the no of Hours to convert into minutes");
            hours=in.nextInt();
            minutes=(hours*60);
            System.out.println("Minutes: " + minutes);
    }
    public void minutestohours()
    {
            System.out.println("Enter the no of Minutes to convert into Hours");
            minutes=in.nextInt();
            hours=minutes/60;
            System.out.println("Hours: " + hours);
    }
    public void hourstoseconds()
    {
            System.out.println("Enter the no of Hours to convert into Seconds");
            hours=in.nextInt();
            seconds=(hours*3600);
            System.out.println("Seconds: " + seconds);
    }
    public void secondstohours()
    {
            System.out.println("Enter the no of Seconds to convert into Hours");
            seconds=in.nextInt();
            hours=seconds/3600;
            System.out.println(seconds+"seconds"+ " is equal to "+hours+"hour");
    }
}
```

Main Class

```
import cc.*;
import dc.*;
```

```
import tc.*;
public class Main
{
    public static void main(String args[])
    {
            CurrencyC obj=new CurrencyC();
            DistanceC obj1=new DistanceC();
            TimeC obj2=new TimeC();

            obj.dollartorupee();
            obj.rupeetodollar();

            obj.eurotorupee();
            obj.rupeetoeuro();

            obj.yentoruppe();
            obj.ruppetoyen();

            obj1.mtokm();
            obj1.kmtom();

            obj1.milestokm();
            obj1.kmtomiles();

            obj2.hourstominutes();
            obj2.minutestohours();

            obj2.hourstoseconds();
            obj2.secondstohours();
    }
}
```

OUTPUT:

```
Enter dollars to convert into Rupees:1
Dollar =1.0 equal to INR=81.83

Enter Rupee to convert into Dollars: 80
Rupee =80.0equal to Dollars=0.977636563607479

Enter the distance in meter :1000
1000.0m is equal to 1.0km

Enter the distance in Kilometre :1
1.0km is equal to 1000.0m

Enter the no of Hours to convert into minutes: 1
Minutes: 60

Enter the no of Minutes to convert into Hours: 60
Hours: 1
```

PROGRAM :7

7. Write a program to generate the resume. Create 2 Java classes Teacher (data: personal information, qualification, experience, achievements) and Student (data: personal inform tion, result, discipline) which implements the java interface Resume with the method biodata().

Aim: Introduction to abstract classes, abstract methods, and Interface in java

PROGRAM:

```
interface Resume
{
        void biodata();
}
class Teacher implements Resume
{
    String name,qualification,achievements;
    float experience;
    public void biodata()
    {
            name="Imran Ulla Khan";
            qualification="M.Tech";
            achievements="Q1 publication";
            experience=14.8f;
            System.out.println("Teacher Resume");
            System.out.println("Name : " +name);
            System.out.println("Qualification : "+qualification);
            System.out.println("Achievements : "+achievements);
            System.out.println("Experience : "+experience);
    }
}
class Student implements Resume
{
    String name,discipline;
    float result;
    public void biodata()
    {
            name="Rahul Sharma";
            result=9.8f;
            discipline="Computer Science and Engineering";
            System.out.println("");
            System.out.println("Student Resume");
            System.out.println("Name : " +name);
            System.out.println("Result : "+result+" cgpa");
            System.out.println("Discipline : "+discipline);
    }
}

public class InterfaceP
{
```

```
    public static void main(String[] args)
    {
            Teacher obj1=new Teacher();
            obj1.biodata();

            Student obj2=new Student();
            obj2.biodata();
    }
}
```

OUTPUT:

```
C:\javasample\JAVALAB>javac InterfaceP.java

C:\javasample\JAVALAB>java InterfaceP
Teacher Resume
Name : Imran Ulla Khan
Qualification : M.Tech
Achievements : Q1 publication
Experience : 14.8

Student Resume
Name : Rahul Sharma
Result : 9.8 cgpa
Discipline : Computer Science and Engineering
```

PROGRAM :8

8. Write a Java program that implements a **multi-thread** application that has three threads. First thread generates a random integer for every 1 second; second thread computesthe square of the number and prints; third thread will print the value of cube of the number.

Aim: Demonstrate creation of threads using Thread class and Runnable interface, multi-threaded programming.

PROGRAM :

```
import java.util.Random;
class Square extends Thread
{
        int x;
        Square(int n)
        {
                x = n;
        }
       public void run()
      {
            int sqr = x * x;
            System.out.println("Square of " + x + " = " + sqr );
       }
}

class Cube extends Thread
{
        int x;
        Cube(int n)
        {
                x = n;
        }
       public void run()
       {
                int cub = x * x * x;
                System.out.println("Cube of " + x + " = " + cub );
       }
}
class Rnumber extends Thread
{
      public void run()
       {
              Random random = new Random();
              for(int i =0; i<5; i++)
               {
                    int randomInteger = random.nextInt(10);
                    System.out.println("Random Integer generated : " + randomInteger);
                    Square s = new Square(randomInteger);
```

```
                    s.start();
                    Cube c = new Cube(randomInteger);
                    c.start();
                    try
                    {
                        Thread.sleep(1000);
                     }
                   catch (InterruptedException ex)
                   {
                        System.out.println(ex);
                   }
               }
        }
}

public class ThreadP
{
        public static void main(String[] args)
        {
                Rnumber n = new Rnumber();
               n.start();

        }
}
```

OUTPUT:

```
C:\javasample\JAVALAB>javac Teg.java

C:\javasample\JAVALAB>java Teg
Random Integer generated : 4
Square of 4 = 16
Cube of 4 = 64
Random Integer generated : 5
Square of 5 = 25
Cube of 5 = 125
Random Integer generated : 9
Square of 9 = 81
Cube of 9 = 729
Random Integer generated : 6
Square of 6 = 36
Cube of 6 = 216
Random Integer generated : 6
Cube of 6 = 216
Square of 6 = 36
```

PROGRAM :9

9. Write a program to perform string operations using ArrayList. Write functions for the following a. Append - add at end b. Insert – add at particular index c. Search d. List all string starts with given letter.

Aim: Introduce java Collections.

```
import java.util.*;
public class ArrayL
{
    ArrayList<String> list=new ArrayList<String>();                //Creating arraylist
    public void arraydisplay()
    {
            list.add("CSE");//Adding object in arraylist
            list.add("ISE");
            list.add("ME");
            System.out.println("ArrayList element are");
            System.out.println(list);
            System.out.println("");
    }
    public void appendatend()
    {
            System.out.println("Enter the element to append at end");
            Scanner scob1=new Scanner(System.in);
            String ele=scob1.next();
            list.add(ele);
            System.out.println(list);
            System.out.println("");
    }
    public void insertatpos()
    {
            System.out.println("Enter the position and element to insert");
            Scanner scob1=new Scanner(System.in);
            int posind=scob1.nextInt();
            String ele=scob1.next();
            list.add(posind,ele);
            System.out.println(list);
            System.out.println("");
    }
    public void searchele()
    {
            System.out.println("Enter the Array element to search");
            Scanner scobj=new Scanner(System.in);
            String arele=scobj.next();
            int in=list.indexOf(arele);
            if(in==-1)
            {
                    System.out.println("Element not found");
            }
```

```
                else
                {
                        System.out.println("Element found at "+in);
                }
        }
        void print()
        {
                Scanner nip=new Scanner(System.in);
                System.out.println("Enter the starting charecter to print strings");
                char inputc=nip.next().charAt(0);
                String strc=Character.toString(inputc);
                System.out.println("String starting with character "+strc);
                for(int i=0;i<list.size();i++)
                {
                        if(list.get(i).startsWith(strc))
                        {
                                System.out.println(list.get(i));
                        }
                }
         }
         public static void main(String args[])
        {
                 ArrayL obj=new ArrayL();
                 obj.arraydisplay();
                 obj.appendatend();
                 obj.insertatpos();
                 obj.searchele();
                 obj.print();
        }
}
```

OUTPUT:

```
C:\javasample\JAVALAB>javac ArrayL.java

C:\javasample\JAVALAB>java ArrayL
ArrayList element are
[CSE, ISE, ME]

Enter the element to append at end
Civil
[CSE, ISE, ME, Civil]

Enter the position and element to insert
1
AIML
[CSE, AIML, ISE, ME, Civil]

Enter the Array element to search
ISE
Element found at 2
Enter the starting charecter to print strings
C
String starting with character C
CSE
Civil
```

PROGRAM :10

10. Write a Java program to read two integers a and b. **Compute** a/b and print, when bis not zero. Raise an exception when b is equal to zero.

Aim: Exce tion h ndling in j , introduction to throw le cl ss, throw, throws, fin lly

```
import java.util.*;
public class TryP
{
    int c;
    void div(int a,int b)
    {
    try
    {
            c=a/b;
            System.out.println("Result="+c);
    }
    catch(ArithmeticException e)
    {
            System.out.println("Cannot divide by zero");
    }
    }
    public static void main(String args[])
    {
            TryP obj=new TryP();
            Scanner in=new Scanner(System.in);
            System.out.println("Enter the values of a and b");
            int no1=in.nextInt();
            int no2=in.nextInt();
            obj.div(no1,no2);
    }
}
```

OUTPUT:

```
C:\javasample\JAVALAB>javac TryP.java

C:\javasample\JAVALAB>java TryP
Enter the values of a and b
10
2
Result=5

C:\javasample\JAVALAB>java TryP
Enter the values of a and b
10
0
Cannot divide by zero
```