

Mechi Multiple Campus

(Tribhuvan University)

Bhadrapur, Jhapa



Lab Report of

Object Oriented Programming in Java (CACS-204)

Faculty of Humanities & Social Sciences

Tribhuvan University

Kritipur, Nepal

Submitted By

Name: Santosh Bhandari

Roll No: 58

Submitted To

Mechi Multiple Campus

Department of Bachelor in Computer

Bhadrapur, Jhapa, Nepal

Certificate from the supervisor

This is to certify that the Lab Report entitled “Object Oriented Programming in Jaya” is an academic work done by “Santosh Bhandari” submitted in the partial fulfillment of the requirements for the degree of **Bachelor of Computer Application** at Faculty of Humanities and Social Science, Tribhuvan University under my guidance and supervision. To the best of my knowledge, the work performed by him in the Lab report is his own creation.

Signature of the Supervisor:-

Name:-

Designation:-

Date:-

Acknowledgement

I would like to express my special thanks of gratitude to my Java teacher “Mr. Sital Prasad Mandal” and to the whole faculty member of humanities and social science who has provided me this opportunity. I am really grateful to sir Sital Prasad Mandal for this kind of support in my report.

We are making this report as it compulsory required by Mechi Multiple Campus. It is compulsory report that should be submitted to the college in order to get practical marks. For this report our Java teacher has motivated us. We are also guided by him to score good marks in practical. He has suggested how to make the report. He was the main source for us to make the report ready for Mechi Multiple Campus. In this report, we the student of BCA were engaged and I would like to express my deep thankful particularly to all of them. We are doing this report to be very helpful for coming days.

Thank You.

Lab No.	EXPERIMENT NAME	Submitted Mark obj.	PAGE NO.
1.	Printing "Welcome to world of Java"	A	1
2.	Printing the Area of Triangle	A	1
3.	Program to check the Number is prime or not	A	1-2
4.	Program to Generate a Ladder of number	A	2
5.	Bitwise Operator Shifting	A	3
6.	For-each loop in Integer and String	A	3-4
7.	Unary Operator	A	4
8.	Program to Create Objects to declare & access value	A	5
9.	Program to Create class with methods	A	6-7
10.	Program to Create parametrized Constructor	A	7
11.	Program to demonstrate constructor Overloading	A	8
12.	Program to create and declared parameter constructor	A	9
13.	Program to demonstrate Compile time polymorphism	A	9-10
14.	Program to demonstrate Run Time polymorphism	A	10-11
15.	OVERRIDING METHODS Example		12
16.	DYNAMIC METHOD Dispatch		12-13
17.	Using Object class Method		13
18.	Using Packages Demo		13-14
19.	Program to Demonstrate Try-Catch		14
20.	Program to Demonstrate Throws		15
21.	Using Different String Methods		15-16
22.	Program to Create Thread and get Status of it		16
23.	Program to Create thread and its priority		17
24.	Program to Demonstrate Hash Map		17
25.	Program to Demonstrate Tree Map		18
26.	Program to Demonstrate Linked Hash Map		18
27.	Program to Demonstrate Array List		19
28.	Program to Demonstrate Linked List		19
29.	Program to Demonstrate Hash Set		20
30.	Program to Demonstrate Tree Set		20
31.	Program to Demonstrate Comparator Interface		21
32.	Program to Demonstrate Vector		22
33.	Program to Demonstrate Stack		22

LAB No.	EXPERIMENT NAME	Submitted Mof's	Page No.
34.	Program to Generate Random Number		23
35.	Program to Demonstrate Hash Table		23
36.	Program to Demonstrate Serialization & Deserialization		24-25
37.	Swing program to perform Calculation of Two Numbers		25-26

1. Write a program to print the text "Welcome to World of Java". Save it with name Welcome.java in your folder.

```
public class Welcome {
    public static void main (String [] args) {
        System.out.println ("Welcome to World of Java .");
    }
}
```

Output

Welcome to World of Java

2. Write a program to print the area of Triangle. Save it with name Area.java in your folder.

```
import java.util.Scanner;
class Area {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter the Base and Height of Triangle : ");
        float b = sc.nextInt ();
        float h = sc.nextInt ();
        System.out.println ("Area = " + (b * h) / 2);
    }
}
```

Output

Enter the Base and Height of Triangle : 7

5

20

Area = 50.0

3. Write a Java program to check the Number is prime or Not.

```
import java.util.*;
class CheckPrime {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter a Number : ");
        int num = sc.nextInt ();
    }
}
```

```

int i, count=0;
for(i=2; i<num; i++) {
    if(num%i==0)
        count++;
        break;
}
if(count==0)
    System.out.println(num+" is a Prime Number.");
else
    System.out.println(num+" is Not a prime Number.");

```

4. Write a Java program to generate a Ladder of Number.

```

class NumberLadder {
    public static void main (String [] args) {
        int n=5, i, j, k;
        for(i=1; i<n; i++) {
            for(j=1; j<i; j++)
                System.out.print(j);
            for(k=j; k>0; k--)
                System.out.print(k);
            System.out.print("\n");
        }
    }
}

```

Output

```

1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1

```

5. Bitwise Operator Shifting

```
import java.util.Scanner;
class BitShift {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the Number:");
        int num = sc.nextInt();
        System.out.println("Before Shifting = " + num + "\nAfter
Shifting = " + (num >>> 2));
    }
}
```

3

Output

Enter the Number:

5

Before Shifting = 5

After Shifting = 2

6. for-each loop (a.) String (b.) Integer

(a.) String

```
import java.util.ArrayList;
import java.util.List;
public class ForEachString {
    public static void main(String[] args) {
        List<String> name = new ArrayList<String>();
        name.add("Ram");
        name.add("Ham");
        name.add("Geeta");
        name.add("Sita");
        for (String n : name)
            System.out.println(name);
    }
}
```

3

Output

Ram
Hari
Geeta
Sita

(b.) Integer

```
public class Integer {
    public static void main(String[] args) {
        int arr[] = {10, 20, 30, 40, 50};
        for(int a : arr)
            System.out.println(a);
    }
}
```

Output

10
20
30
40
50

7. Unary Operator Example: Complement (\sim)

```
import java.util.Scanner;
class Unaryoperator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a Number: ");
        int num = sc.nextInt();
        System.out.println("Before Complement = " + num + "\nAfter Complement = " + (~num));
    }
}
```

Output

Enter a number:

10

Before Complement = 10

After Complement = -11

~~S. I. D.~~
06-05-29

8. Write a program to create a class Student with data "name, city and age" along with method printData to display the data. Create the two objects s1, s2 to declare and access the values.

```
class Student {
    String name, city;
    int age;
    void printData() {
        System.out.println("Name = " + name + "\nCity = " + city + "\n"
                           "Age = " + age);
    }
}
```

```
class StudentDetails {
    public static void main(String[] args) {
        Student s1 = new Student();
        Student s2 = new Student();
        s1.name = "Harry";
        s1.city = "Birtamode";
        s1.age = 20;
        s2.name = "Anish";
        s2.city = "Bhadrapur";
        s2.age = 28;
        s1.printData();
        s2.printData();
    }
}
```

Output

```
Name = Harry
City = Birtamode
Age = 20
Name = Anish
City = Bhadrapur
Age = 28
```

- 90 Write a program to create a class Student2 along with two method getData(), printData() to get the value through argument and display the data in printData. Create the two objects S1,S2 to declare and access the values from class STtest.

```

import java.util.Scanner;
class Student2 {
    private String name, address;
    private int roll;
    void getData(int r, String n, String a) {
        roll = r;
        name = n;
        address = a;
    }
    void printData() {
        System.out.println("Roll. No. = " + roll + "\nName = " + name + "\n"
            + address);
    }
}
class STtest {
    public static void main (String args[]) {
        Scanner sc = new Scanner(System.in);
        Student2 s1 = new Student2();
        Student2 s2 = new Student2();
        System.out.println("Enter Roll. No., Name and Address of
Two Student : ");
        int r = sc.nextInt();
        String n = sc.next(), a = sc.next();
        s1.getData(r, n, a);
        r = sc.nextInt();
        n = sc.next();
        a = sc.next();
        s2.getData(r, n, a);
        s1.printData();
        s2.printData();
    }
}

```

Output

Enter the Roll.No., Name and Address of Two Students
5
Kiran

Birtamode

12

Ayush

Bhadrapur

Roll. No. = 5

Name = kiran

Address = Birtamode

Roll. No. = 12

Name = Ayush

Address = Bhadrapur

10.

WAP using parameterized Employee constructor with two parameters id and name. While creating the objects obj1 and obj2 passed two arguments so that this constructor get invoked after creation of obj1 and obj2.

```
class Employee {
    int id;
    String name;
    Employee(int id, String name) {
        this.id = id;
        this.name = name;
    }
    void Display() {
        System.out.println("ID = "+id+" Name = "+name);
    }
}
```

```
class EmployeeConstructor
```

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

```
Employee obj1 = new Employee(103, "Ansh");
```

```
Employee obj2 = new Employee(205, "Manoj");
```

```
obj1.Display();
```

```
obj2.Display();
```

```
3
```

Output

ID = 103

Name = Ansh

ID = 205

Name = Manoj

11. Write a program in Java to demonstrate the method and constructor overloading.

class Cons

int x, y;

Cons() {

}

Cons(int a, int b) {

x = a;

y = b;

}

void Sum(int x, int y) {

System.out.println("Sum of Integer = " + (x+y));

void Sum(float x, float y) {

System.out.println("Sum of Float = " + (x+y));

void printData() {

System.out.println("X = " + x + " Y = " + y);

}

class OverloadingDemo {

public static void main(String[] args) {

Cons c1 = new Cons();

Cons c2 = new Cons(10, 20);

c2.printData();

c1.Sum(10, 20);

c2.Sum(3.2f, 4.3f);

}

}

Output

X = 10

Y = 20

Sum of Integer = 30

Sum of Float = 7.5

12. Write a program in Java to Create a class Bird also declares the different parameterized constructor to display the name of Birds.

```

class Bird {
    Bird(String name) {
        System.out.println("Name = " + name);
    }
    Bird(String name, float weight) {
        System.out.println("Name = " + name + " Weight = " + weight);
    }
    Bird(String name, int age, float weight) {
        System.out.println("Name = " + name + " Age = " + age + " Weight = " + weight);
    }
}

```

```

class DifferentParametersDemo {
    public static void main(String args[]) {
        Bird b1 = new Bird("Sparrow");
        Bird b2 = new Bird("Dove", 0.5f);
        Bird b3 = new Bird("Parrot", 2, 0.9f);
    }
}

```

Output

```

Name = Sparrow
Name = Dove
Weight = 0.5
Name = Parrot
Age = 2
Weight = 0.9

```

13. Demonstrate Simpler calculator using Java. Here we have two definitions of the same method add() which add method would be called is determined by the parameter list at the compile time.

```

class Calc {
    void add(int x, int y) {
        System.out.println("Integer Addition = " + (x+y));
    }
    void add(float x, float y) {
        System.out.println("Float Addition = " + (x+y));
    }
}

```

```

class CompleteTimeDemo {
    public static void main(String[] args) {
        Calc c1 = new Calc();
        c1.add(15, 25);
        c1.add(5.62f, 63.31f);
    }
}

```

3

Output

Integer Addition = 40

Float Addition = 68.93

14. Demonstrate Simple Shape Using Java. Here we have three definitions of the same method draw() which different draw method, would be called is determined by their respected shape at the run time.

```
class Shape {
```

```
    void Draw() {
        System.out.println("Shape");
    }
}
```

```
class Circle extends Shape {
```

```
    void Draw() {
        System.out.println("Circle");
    }
}
```

```
class Triangle extends Shape {
```

```
    void Draw() {
        System.out.println("Triangle");
    }
}
```

```
class Rectangle extends Shape {
```

```
    void Draw() {
        System.out.println("Rectangle");
    }
}
```

```
class RunTimeDemo {
```

```
    public static void main(String[] args) {
        Shape c = new Circle();
        Shape t = new Triangle();
        Shape r = new Rectangle();
    }
}
```

```
        Shape c = new Circle();
```

```
        Shape t = new Triangle();
```

```
        Shape r = new Rectangle();
```

```
c.Draw();  
t.Draw();  
r.Draw();
```

{
3**Output**

Circle
Triangle
Rectangle

Sides
2-0 & -79

15. Overriding Methods Example

```

class Parent {
    void Disp() {
        System.out.println("Parent Class");
    }
}

class Child extends Parent {
    void Disp() {
        System.out.println("Child Class");
    }

    void Message() {
        Disp();
    }
}

class OverridingExample {
    public static void main(String args[]) {
        Child obj = new Child();
        obj.Message();
    }
}

```

Output

Child class

16. Dynamic Method Dispatch

```

class Parent {
    void Output() {
        System.out.println("Parent");
    }
}

class Child extends Parent {
    void Output() {
        System.out.println("Child");
    }

    void Display() {
        System.out.println("This is new Method");
    }
}

```

```

class DMDEExample{
    public static void main (String args[]){
        Parent obj = new Child();
        obj.Display();
    }
}

```

Output

Compile Error

17. Using Object class Method

```

class Student{
}

class ObjectClassMethod{
    public static void main (String args[]){
        Student obj = new Student();
        System.out.println(obj.getClass());
    }
}

```

Output

class Student

18. Using Packages Demo

```

package info;
public class Name{
    public void Name(String n){
        System.out.println("Hey, "+n);
    }
}

```

```

import java.util.Scanner;
import info.Name;
class packageDemo{
    public static void main (String args[]){
}

```

```

Name name = new Name();
Scanner sc = new Scanner(System.in);
System.out.println("What is your Name : ");
String n = sc.nextLine();
name.Name(n);

```

3

Output

What is your Name :

Santosh

Hey, Santosh

19. WAP to Demonstrate try-catch.

```

import java.util.Scanner;
class TryCatchDemo {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Two Numbers :- ");
        int a = sc.nextInt(), b = sc.nextInt();
        try {
            System.out.println("Division = " + (a/b));
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}

```

3

Output

Enter Two Numbers :-

10

2

Division = 5

20. WAP to Demonstrate Throws.

```

import java.util.Scanner;
class ThrowsDemo {
    public static void main(String args[]) throws ArithmeticException {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Two Numbers: ");
        int a = sc.nextInt(), b = sc.nextInt();
        System.out.println(a + " / " + b + " = " + (a/b));
    }
}

```

Output

Enter Two Numbers:

5

0

Exception in thread "main" java.lang.ArithmeticException: / by zero
at ThrowsDemo.main(ThrowsDemo.java:7)

21. WAP to Demonstrate different String Methods.

```

class StringMethods {
    public static void main(String args[]) {
        String str1 = "Hello ", str2 = "World";
        System.out.println(str1.concat(str2));
        System.out.println(str1.length());
        System.out.println(str2.indexOf("l"));
        System.out.println(str1.toLowerCase());
        System.out.println(str2.toUpperCase());
        System.out.println(str1.equals(str2));
        System.out.println(str1.replaceAll("l", "d"));
    }
}

```

3

3

Output

Hello World

6

3

hello
WORLD
false
Hellooo

22. Write a program to create thread and get the status of the thread.

```
class ThreadState extends Thread {
    public void run() {
        System.out.println("Thread is Running");
    }
    public static void main(String args[]) {
        ThreadState t1 = new ThreadState();
        ThreadState t2 = new ThreadState();
        System.out.println("t1 state: " + t1.getState());
        System.out.println("t2 state: " + t2.getState());
        t1.start();
        System.out.println("t1 state: " + t1.getState());
        t2.start();
        System.out.println("t2 state: " + t2.getState());
    }
}
```

3

Output

t1 state: New
 t2 state: New
 t1 state: Runnable
 Thread is Running
 t2 state: Runnable
 Thread is Running

23. WAP to Create Thread and demonstrate thread priority.

```
class ThreadPriority extends Thread {
    public void run() {
        System.out.println(Thread.currentThread() + " Thread is running and
                           priority is " + Thread.currentThread().getPriority());
    }
    public static void main(String args[]) {
        ThreadPriority t1 = new ThreadPriority();
        ThreadPriority t2 = new ThreadPriority();
        t1.setPriority(Thread.MAX_PRIORITY);
        t2.setPriority(Thread.MIN_PRIORITY);
        t2.start();
        t1.start();
    }
}
```

3

Output

Thread[#22, Thread-0, 10, main] Thread is running and priority is 10

Thread[#23, Thread-1, 1, main] Thread is running and priority is 1

24. WAP to demonstrate Hash Map in Java

```
import java.util.HashMap;
class HashMapDemo {
    public static void main(String args[]) {
        HashMap<Integer, String> map = new HashMap<Integer, String>();
        map.put(1, "C Programming");
        map.put(2, "Java Programming");
        map.put(3, "PHP Programming");
        map.put(4, "C++ Programming");
        System.out.println(map);
    }
}
```

3

Output

{1=C programming, 2=Java programming, 3=PHP programming, 4=C++
Programming}

25. WAP to Implement Tree map.

```

import java.util.TreeMap;
import java.util.Map;
class TreeMapDemo {
    public static void main(String args[]) {
        Map<Integer, String> map = new TreeMap<Integer, String>();
        map.put(1, "C");
        map.put(2, "Java");
        map.put(3, "PHP");
        map.put(4, "C++");
        System.out.println(map);
        System.out.println(map.get(3));
    }
}

```

Output

{1=C, 2=Java, 3=PHP, 4=C++}
PHP

26. Write a program to demonstrate the use of linked Hash Map.

```

import java.util.LinkedHashMap;
import java.util.Map;
class LinkedHashMapDemo {
    public static void main(String args[]) {
        Map<Integer, String> map = new LinkedHashMap<Integer, String>();
        map.put(1, "C");
        map.put(2, "Java");
        map.put(3, "PHP");
        map.put(4, "C++");
        System.out.println(map);
        System.out.println(map.get(3));
    }
}

```

Output

{1=C, 2=Java, 3=PHP, 4=C++}
PHP

27. Write a program to demonstrate Array list.

```
import java.util.ArrayList;
import java.util.List;
class ListDemo {
    public static void main(String args[]) {
        List list = new ArrayList();
        list.add("C++ Programming");
        list.add("C Programming");
        list.add("Java Programming");
        list.add("Python Programming");
        System.out.println(list);
    }
}
```

3

Output

[C++ Programming, C Programming, Java Programming, Python Programming]

28. WAP to Demonstrate the linked list.

```
import java.util.LinkedList;
class LinkedListDemo {
    public static void main(String args[]) {
        LinkedList list = new LinkedList();
        list.add("Java");
        list.add("C");
        list.add("C++");
        list.add("Python");
        list.add("Rust");
        System.out.println(list);
    }
}
```

3

Output

[Java, C, C++, Python, Rust]

29. Write a program to demonstrate HashSet

```
import java.util.HashSet;
class HashSetDemo {
    public static void main(String args[]) {
        HashSet<String> set = new HashSet<String>();
        set.add("Maths");
        set.add("Science");
        set.add("Nepali");
        set.add("English");
        set.add("Social");
        System.out.println(set);
    }
}
```

3

Output

[Social, Maths, English, Science, Nepali]

30. Write a program to demonstrate Tree Set in Java.

```
import java.util.TreeSet;
import java.util.Set;
class TreeSetDemo {
    public static void main(String args[]) {
        Set set = new TreeSet();
        set.add(250);
        set.add(150);
        set.add(15);
        set.add(168);
        set.add(300);
        System.out.println(set);
    }
}
```

3

Output

[15, 150, 168, 250, 300]

31. Write a program to Demonstrate Comparator Interface.

```
import java.util.*;
class Student {
    int rno, age;
    String name;
    Student (int r, String n, int a) {
        rno = r;
        name = n;
        age = a;
    }
}
```

3

```
class RollnoComparator implements Comparator<Student> {
    public int compare(Student s1, Student s2) {
        if (s1.rno > s2.rno)
            return 1;
        else if (s1.rno < s2.rno)
            return -1;
        else
            return 0;
    }
}
```

3

```
class ComparatorInterfaceDemo {
    public static void main (String args[]) {
        ArrayList<Student> llist = new ArrayList<Student>();
        llist.add(new Student(5, "Sita", 22));
        llist.add(new Student(1, "Ram", 20));
        llist.add(new Student(3, "Geeta", 19));
        llist.add(new Student(32, "Shyam", 25));
        llist.add(new Student(10, "Ravi", 23));
        Collections.sort(llist, new RollnoComparator());
        for (Student s : llist)
            System.out.println(s.rno + " " + s.name + " " + s.age);
    }
}
```

3

Output

1	Ram	20
3	Geeta	19
5	Sita	22
10	Ravi	23
32	Shyam	25

32. Write a program to Demonstrate Vector

```
import java.util.Vector;
class VectorDemo {
    public static void main (String args[]){
        Vector lot = new Vector();
        lot.add(10);
        lot.add(50);
        lot.add(0);
        lot.add(15);
        lot.addElement(35);
        lot.insertElementAt(17,3);
        System.out.println(lot);
    }
}
```

3

Output

[15, 10, 50, 17, 35]

33. Write a program to Demonstrate Stack

```
import java.util.Stack;
class StackDemo {
    public static void main (String args[]){
        Stack stack = new Stack();
        stack.push(102);
        stack.push(103);
        stack.push(105);
        stack.push(115);
        System.out.println(stack);
        System.out.println(stack.pop());
        System.out.println(stack);
        System.out.println(stack.pop());
        System.out.println(stack.pop());
        System.out.println(stack);
    }
}
```

3

3

Output

[102, 103, 105, 115]

115
[102, 103, 105]

105

103

[102]

34. Write a program to generate random number in Java.

```
import java.util.Random;
class RandomNumberGenerator {
    public static void main (String args[]) {
        Random n = new Random ();
        int num = n.nextInt(500);
        System.out.println (num);
    }
}
```

3

Output

34 // You will get Unique Number in every execution

35. Write a program to demonstrate hash table.

```
import java.util.Hashtable;
class HashTableDemo {
    public static void main (String args[]) {
        Hashtable d = new Hashtable ();
        d.put(1, "C");
        d.put(10, "C++");
        d.put(31, "Python");
        d.put(5, "Java");
        System.out.println (d);
    }
}
```

3

Output

{10=C++, 31=Python, 5=Java, 1=C}

360

Write a program to Demonstrate Serialization and Deserialization.

```

import java.io.*;
import java.util.Scanner;
class Data implements Serializable {
    int rno;
    String name;
    public Data(int n, String na) {
        rno = n;
        name = na;
    }
}
class SerializationDeserialization {
    public static void main (String args[]) {
        try {
            Scanner sc = new Scanner(System.in);
            System.out.println ("Enter a Roll No. and Name:");
            int r = sc.nextInt();
            String n = sc.next();
            Data d1 = new Data(r, n);
            FileOutputStream out = new FileOutputStream ("data.txt");
            ObjectOutputStream obj = new ObjectOutputStream (out);
            obj.writeObject (d1);
            obj.flush();
            obj.close();
            FileInputStream in = new FileInputStream ("data.txt");
            ObjectInputStream objIn = new ObjectInputStream (in);
            Data d = (Data) objIn.readObject();
            System.out.println ("Roll No.: " + d.rno + " Name: " + d.name);
            objIn.close();
        } catch (Exception e) {
            System.out.println (e);
        }
    }
}

```

3

3

Output

Enter a Roll No. and Name:

25

Anish

Roll No: 25

Name: Anish

37. Write a Swing Program to ask the two Number and Sum, Sub, Multiply and Divide those Number.

```

import java.awt.*;
import java.swing.*;
import java.awt.event.*;
class CalculationUsingSwing {
    static JTextField t1, t2, t3, t4, t5, t6;
    public static void main (String args[]) {
        JFrame f = new JFrame ("Calculation");
        t1 = new JTextField ();
        t1.setBounds (100, 50, 200, 30);
        t2 = new JTextField ();
        t2.setBounds (150, 100, 200, 30);
        t3 = new JTextField ();
        t3.setBounds (150, 150, 200, 30);
        t4 = new JTextField ();
        t4.setBounds (150, 200, 200, 30);
        t5 = new JTextField ();
        t5.setBounds (150, 250, 200, 30);
        t6 = new JTextField ();
        t6.setBounds (150, 300, 200, 30);
        f.add (t1);
        f.add (t2);
        f.add (t3);
        f.add (t4);
        f.add (t5);
        f.add (t6);
        JLabel n1, m2, n3, n4, n5, n6;
        n1 = new JLabel ("First Number");
        n1.setBounds (50, 50, 100, 30);
    }
}

```

```

n2=new JLabel ("Second Number");
n2.setBounds (50,100,100,30);
n3=new JLabel ("Addition");
n3.setBounds (50,200,100,30);
n4=new JLabel ("Subtraction");
n4.setBounds (50,250,100,30);
n5=new JLabel ("Multiplication");
n5.setBounds (50,300,100,30);
n6=new JLabel ("Division");
n6.setBounds (50,350,100,30);
f.add(n1); f.add(n2); f.add(n3); f.add(n4); f.add(n5);
f.add(n6);

JButton btn = new JButton ("Calculate"),
btn.setBounds (100,150,90,30),
btn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        t3.setText(Integer.toString(Integer.parseInt(t1.getText()) +
        Integer.parseInt(t2.getText())));
        t4.setText(Integer.toString(Integer.parseInt(t1.getText())
        - Integer.parseInt(t2.getText())));
        t5.setText(Integer.toString(Integer.parseInt(t1.getText()) *
        Integer.parseInt(t2.getText())));
        t6.setText(Integer.toString(Integer.parseInt(t1.getText())
        /Integer.parseInt(t2.getText())));
    }
});
f.add(btn);
f.setSize(400,450);
f.setLayout(null);
f.setVisible(true);

```

3

3