Experiment No. 01

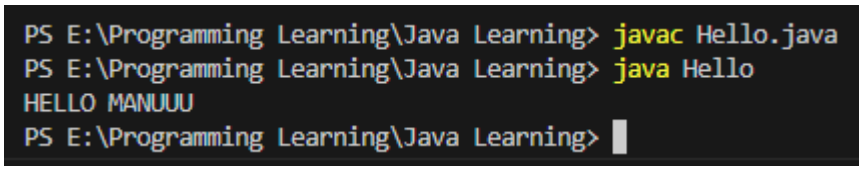
مقصد:

اسکرین پر "Hello! MANUU " پرنٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
class Hello
{
       public static void main(String a[])
      {
             System.out.println("HELLO MANUUU");
       }
}
```

نتیجہ:



Experiment No. 02

مقصد:

sum(), mul(), sub(),div() فنکشنز کا استعمال کرتے ہوئے Arithmetical Operation کو ڈیمونسٹریٹ کرنیکے لئے ایک جاوا کا پروگرام لکھئیے.

پروگرام:

```
import java.util.Scanner;

public class OperationFunction {
  public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    System.out.println("Enter any two positive integer numbers:");

    int p = in.nextInt();
    int q = in.nextInt();

    System.out.println("\nSUM        " + p + " + " + q + " = " + Sum(p, q));
    System.out.println("DIFFERENCE  " + p + " - " + q + " = " + Sub(p, q));
    System.out.println("PRODUCT     " + p + " * " + q + " = " + Mul(p, q));
    System.out.println("QUOTIENT    " + p + " / " + q + " = " + Div(p, q));
    System.out.println("MODULUS     " + p + " % " + q + " = " + Mod(p, q));
  }


public static int Sum(int x, int y) {
    return x + y;
  }

  public static int Sub(int x, int y) {
    return x - y;
  }

  public static int Mul(int x, int y) {
    return x * y;
  }

  public static float Div(int x, int y) {
    return x / y;
  }

  public static int Mod(int x, int y) {
    return x % y;
  }
}
```

نتیجہ:

```
PS E:\Programming Learning\Danish Java> java OperationFunction
Enter any two positive integer numbers:
15 15

SUM         15 + 15 = 30
DIFFERENCE  15 - 15 = 0
PRODUCT     15 * 15 = 225
QUOTIENT    15 / 15 = 1.0
MODULUS     15 % 15 = 0
PS E:\Programming Learning\Danish Java>
```

Experiment No. 03

مقصد:

مختلف شیپس کے عریاں جیسے triangle، rectangle اور square کا حساب کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
import java.util.Scanner;

public class Area {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("1. Calculate the area of a triangle");
        System.out.println("2. Calculate the area of a rectangle");


System.out.println("3. Calculate the area of a square");
        System.out.print("Enter your choice (1-3): ");
        int choice = scanner.nextInt();

        switch (choice) {
            case 1:
                System.out.print("Enter the base length of the triangle: ");
                double base = scanner.nextDouble();
                System.out.print("Enter the height of the triangle: ");
                double height = scanner.nextDouble();
                double triangleArea = 0.5 * base * height;
                System.out.println("The area of the triangle is: " + triangleArea);
                break;

            case 2:
                System.out.print("Enter the length of the rectangle: ");
                double length = scanner.nextDouble();
                System.out.print("Enter the width of the rectangle: ");
                double width = scanner.nextDouble();
                double rectangleArea = length * width;
                System.out.println("The area of the rectangle is: " + rectangleArea);
                break;

            case 3:
                System.out.print("Enter the side length of the square: ");
                double side = scanner.nextDouble();
                double squareArea = side * side;
                System.out.println("The area of the square is: " + squareArea);
                break;
```

```
            default:
                System.out.println("Invalid choice!");
        }

        scanner.close();
    }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> javac Area.java
PS E:\Programming Learning\JavaByDanish> java Area
1. Calculate the area of a triangle
2. Calculate the area of a rectangle
3. Calculate the area of a square
Enter your choice (1-3): 2
Enter the length of the rectangle: 5
Enter the width of the rectangle: 5
The area of the rectangle is: 25.0
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 04

مقصد:

تین نمبروں میں سے سب سے بڑا نمبر تلاش کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
import java.util.Scanner;

public class BiggestNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the first number: ");
        int num1 = scanner.nextInt();

        System.out.print("Enter the second number: ");
        int num2 = scanner.nextInt();

        System.out.print("Enter the third number: ");
        int num3 = scanner.nextInt();

        int max = num1;

        if (num2 > max) {
            max = num2;
        }

        if (num3 > max) {
            max = num3;
        }
        System.out.println("The biggest number is: " + max);
    }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> javac BiggestNumber.java
PS E:\Programming Learning\JavaByDanish> java BiggestNumber
Enter the first number: 20
Enter the second number: 30
Enter the third number: 5
The biggest number is: 30
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 05

مقصد:

سٹرنگ کلاس کے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
public class StringClass {
  public static void main(String[] args) {
      String str1 = "Hello,";
      String str2 = " world!";

      String result = str1 + str2;
      System.out.println("Concatenated string: " + result);

      int length = result.length();
      System.out.println("Length of the string: " + length);

      String substring = result.substring(7, 12);
      System.out.println("Substring: " + substring);

      char character = result.charAt(0);
      System.out.println("Character at index 0: " + character);

      int index = result.indexOf("world");
      System.out.println("Index of 'world': " + index);

      String replaced = result.replace("o", "x");
      System.out.println("String with replaced characters: " + replaced);

      String uppercase = result.toUpperCase();
      System.out.println("Uppercase string: " + uppercase);

      String lowercase = result.toLowerCase();
      System.out.println("Lowercase string: " + lowercase);

      boolean startsWith = result.startsWith("Hello");
      System.out.println("Starts with 'Hello': " + startsWith);

      boolean endsWith = result.endsWith("world!");
      System.out.println("Ends with 'world!': " + endsWith);

      boolean contains = result.contains("lo,");
      System.out.println("Contains 'lo,': " + contains);
  }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> javac StringClass.java
PS E:\Programming Learning\JavaByDanish> java StringClass
Concatenated string: Hello, world!
Length of the string: 13
Substring: world
Character at index 0: H
Index of 'world': 7
String with replaced characters: Hellx, wxrld!
Uppercase string: HELLO, WORLD!
Lowercase string: hello, world!
Starts with 'Hello': true
Ends with 'world!': true
Contains 'lo,': true
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 06

مقصد:

ایک جاوا پروگرام لکھئیے جس میں Student کے دو آبجیکٹس ہونا چاہئے اور insert() میتھڈ کے ذریعہ اس آبجیکٹس کو values initialize کرئے. آخر میں display() میتھڈ کے ذریعہ آبجیکٹس کے data کو display کرایے.

پروگرام:

```
class Student {
    int rollNo;
    String name;

    // Method to insert student data
    void insert(int rollNo, String name) {
        this.rollNo = rollNo;
        this.name = name;
    }

    // Method to display student data
    void display() {
        System.out.println("Student Roll No: " + rollNo);
        System.out.println("Student Name: " + name);
    }
}

public class StudentDisplay  {
    public static void main(String[] args) {
        // Create two Student objects
        Student student1 = new Student();
        Student student2 = new Student();

        // Insert data for student1
        student1.insert(1, "John Doe");

        // Insert data for student2
        student2.insert(2, "Jane Smith");

        // Display student information
        System.out.println("Student 1 Details:");
        student1.display();

        System.out.println("\nStudent 2 Details:");
        student2.display();
    }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> java StudentDisplay
Student 1 Details:
Student Roll No: 1
Student Name: John Doe

Student 2 Details:
Student Roll No: 2
Student Name: Jane Smith
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 07

مقصد:

Command line argument کا استعمال کرتے ہوئےتین نمبروں میں سے سب سے بڑا نمبر تلاش کرنے کے لئے جاوا کا پروگرام لکھیئے.

پروگرام:

```
public class BigNumCmndLine {
    public static void main(String[] args) {
        if (args.length == 0) {
            System.out.println("No numbers provided.");
            return;
        }

        // Assume the first number is the largest
        int max = Integer.parseInt(args[0]);

        for (int i = 1; i < args.length; i++) {
            int current = Integer.parseInt(args[i]);
            if (current > max) {
                max = current;
            }
        }

        System.out.println("The biggest number is: " + max);
    }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> java BigNumCmndLine 40 50 60 95
The biggest number is: 95
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 08

مقصد:

this keyword اور this() method کا طریقہ استعمال کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
class Person {
    private String name;

    public Person(String name) {
        this.name = name;
    }

    public void sayHello() {
        System.out.println("Hello, my name is " + this.name + ".");
    }
}

public class ThisKeyword {
    public static void main(String[] args) {
        Person person = new Person("John");
        person.sayHello();
    }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> javac ThisKeyword.java
PS E:\Programming Learning\JavaByDanish> java ThisKeyword
Hello, my name is John.
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 09

مقصد:

Constructor overloading کو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
class Rectangle {
    private int length;
    private int width;

    public Rectangle() {
        length = 0;
        width = 0;
    }

    public Rectangle(int side) {
        length = side;
        width = side;
    }

    public Rectangle(int length, int width) {
        this.length = length;
        this.width = width;
    }

    public int calculateArea() {
        return length * width;
    }
}

public class ConstOver {
    public static void main(String[] args) {
        Rectangle square = new Rectangle(5);
        System.out.println("Area of square: " + square.calculateArea());

        Rectangle rectangle = new Rectangle(4, 6);
        System.out.println("Area of rectangle: " + rectangle.calculateArea());
    }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> javac ConstOver.java
PS E:\Programming Learning\JavaByDanish> java ConstOver
Area of square: 25
Area of rectangle: 24
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 10

مقصد:

مختلف ٹائپ کاسٹنگ کو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

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

        int myInt = 100;
        long myLong = myInt;

        System.out.println("Implicit Casting:");
        System.out.println("myInt: " + myInt);
        System.out.println("myLong: " + myLong);

        // Explicit Type Casting
        double myDouble = 123.456;
        int myNewInt = (int) myDouble;

        System.out.println("\nExplicit Casting:");
        System.out.println("myDouble: " + myDouble);
        System.out.println("myNewInt: " + myNewInt);
    }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> javac TypeCasting.java
PS E:\Programming Learning\JavaByDanish> java TypeCasting
Implicit Casting:
myInt: 100
myLong: 100

Explicit Casting:
myDouble: 123.456
myNewInt: 123
```

Experiment No. 11

مقصد:

1-D array کا استعمال کرتے ہوئے رن ٹائم میں عدد نمبروں کو پڑھنے اور ڈسپلے کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

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

        int[] numbers = {5, 10, 15, 20, 25};

        System.out.println("Random Numbers:");
        for (int i = 0; i < numbers.length; i++) {
            System.out.println(numbers[i]);
        }
    }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> java  OneDArr
Random Numbers:
5
10
15
20
25
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 12

مقصد:

ایک integer array کو ascending اور descending کے میں sort کرنے کے لئے جاوا کا پروگرام لکھیئے.

پروگرام:

```
import java.util.Arrays;

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

    int[] numbers = { 10, 5, 8, 3, 1 };

    // Ascending Order Sorting
    Arrays.sort(numbers);
    System.out.println("Ascending Order:");
    for (int i = 0; i < numbers.length; i++) {
      System.out.print(numbers[i] + " ");
    }
    System.out.println();

    // Descending Order Sorting
    for (int i = 0; i < numbers.length / 2; i++) {
      int temp = numbers[i];
      numbers[i] = numbers[numbers.length - 1 - i];
      numbers[numbers.length - 1 - i] = temp;
    }
    System.out.println("Descending Order:");
    for (int i = 0; i < numbers.length; i++) {
      System.out.print(numbers[i] + " ");
    }
  }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> java AscDsc
Ascending Order:
1 3 5 8 10
Descending Order:
10 8 5 3 1
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 13

مقصد:

کوئی بی ناموں کی دی گئی فہرست کو ascending order میں sort کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
import java.util.Arrays;

public class NameAsc {

    public static void main(String[] args) {
        String[] names = {"Danish", "Mohtesham","Umar", "Shaquib", "Hamd", "Pasha"};

        // Ascending Order Sorting
        Arrays.sort(names);
        System.out.println("Ascending Order:");
        for (int i = 0; i < names.length; i++) {
            System.out.println(names[i]);
        }
    }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> java NameAsc
Ascending Order:
Danish
Hamd
Mohtesham
Pasha
Shaquib
Umar
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 14

مقصد:

دو matrices کی addition کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
public class MatrixAddition {
  public static void main(String[] args) {
    int[][] matrix1 = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };
    int[][] matrix2 = { { 9, 8, 7 }, { 6, 5, 4 }, { 3, 2, 1 } };

    int rows = matrix1.length;
    int columns = matrix1[0].length;

    int[][] sumMatrix = new int[rows][columns];

    // Matrix Addition
    for (int i = 0; i < rows; i++) {
      for (int j = 0; j < columns; j++) {
        sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];

      }
    }

    // Display the Sum Matrix
    System.out.println("Sum Matrix:");
    for (int i = 0; i < rows; i++) {
      for (int j = 0; j < columns; j++) {
        System.out.print(sumMatrix[i][j] + " ");
      }
      System.out.println();
    }
  }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> java MatrixAddition
Sum Matrix:
10 10 10
10 10 10
10 10 10
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 15

مقصد:

دو matrices کی multiplication کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
public class MatrixMultiplication {
  public static void main(String[] args) {
    int[][] matrix1 = { { 1, 2, 3 }, { 4, 5, 6 } };
    int[][] matrix2 = { { 7, 8 }, { 9, 10 }, { 11, 12 } };

    int rows1 = matrix1.length;
    int columns1 = matrix1[0].length;
    int rows2 = matrix2.length;
    int columns2 = matrix2[0].length;

    if (columns1 != rows2) {
      System.out.println("Matrix multiplication is not possible.");
      return;
    }

    int[][] productMatrix = new int[rows1][columns2];

    // Matrix Multiplication
    for (int i = 0; i < rows1; i++) {
      for (int j = 0; j < columns2; j++) {
        for (int k = 0; k < columns1; k++) {
          productMatrix[i][j] += matrix1[i][k] * matrix2[k][j];
        }
      }
    }

    // Display the Product Matrix
    System.out.println("Product Matrix:");
    for (int i = 0; i < rows1; i++) {
      for (int j = 0; j < columns2; j++) {
        System.out.print(productMatrix[i][j] + " ");
      }
      System.out.println();
    }
  }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> java MatrixMultiplication
Product Matrix:
58 64
139 154
```

Experiment No. 16

مقصد:

Inheritance کے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
// Superclass (پدر کلاس)
class Vehicle {
  String brand;

  void drive() {
    System.out.println("Driving a vehicle");
  }
}

// Subclass (بیٹا کلاس)
class Car extends Vehicle {
  int numberOfDoors;

  void accelerate() {
    System.out.println("Accelerating the car");
  }
}

public class Inheritance {
  public static void main(String[] args) {
    // Create an object of the Car class
    Car myCar = new Car();

    // Access inherited member from the superclass
    myCar.brand = "Toyota";
    System.out.println("Brand: " + myCar.brand);

    // Access member of the subclass
    myCar.numberOfDoors = 4;
    System.out.println("Number of Doors: " +

        myCar.numberOfDoors);

    // Call inherited method from the superclass
    myCar.drive();

    // Call method of the subclass
    myCar.accelerate();
  }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> java Inheritance
Brand: Toyota
Number of Doors: 4
Driving a vehicle
Accelerating the car
```

Experiment No. 17

مقصد:

درج ذیل دی گئی معلومات کوScanner کلاس کے آبجیکٹ کی مدد سے read کرے اور نتائج دکھانے کے لئے جاوا کا پروگرام لکھئیے "EMPLOYEE NAME, AGE, BLOOD GROUP and HEIGHT" :

پروگرام:

```
import java.util.Scanner;

public class EmpInfo {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter Employee Name: ");
    String name = scanner.nextLine();

    System.out.print("Enter Employee Age: ");
    int age = scanner.nextInt();
    scanner.nextLine(); // Consume the remaining newline character

    System.out.print("Enter Employee Blood Group: ");
    String bloodGroup = scanner.nextLine();

    System.out.print("Enter Employee Height: ");
    double height = scanner.nextDouble();

    // Display the Employee Information
    System.out.println("Employee Name: " + name);
    System.out.println("Employee Age: " + age);
    System.out.println("Employee Blood Group: " + bloodGroup);
    System.out.println("Employee Height: " + height);

    scanner.close();
  }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> java EmpInfo
Enter Employee Name: Danish Kamal
Enter Employee Age: 23
Enter Employee Blood Group: A+
Enter Employee Height: 6 feet
Employee Name: Danish Kamal
Employee Age: 23
Employee Blood Group: A+
Employee Height: 6.0
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 18

مقصد:

Abstract class اور abstract method کے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
// Abstract class (غیر مکمل کلاس)
abstract class Shape {
  String color;

  // Abstract method (غیر مکمل میتھڈ)
  abstract double calculateArea();

  void setColor(String color) {
    this.color = color;
  }
}

// Concrete class (مکمل کلاس)

class Rectangle extends Shape {
  double length;
  double width;

  Rectangle(double length, double width) {
    this.length = length;
    this.width = width;
  }

  // Implementation of abstract method (غیر مکمل میتھڈ کی تنسیخ)
  double calculateArea() {
    return length * width;
  }
}

public class Abstract {
  public static void main(String[] args) {
    // Create an object of the Rectangle class
    Rectangle rectangle = new Rectangle(5, 7);
    rectangle.setColor("Blue");

    // Call the abstract method
    double area = rectangle.calculateArea();

    // Display the result
    System.out.println("Rectangle Area: " + area);
```

```
    System.out.println("Rectangle Color: " + rectangle.color);
  }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> java Abstract
Rectangle Area: 35.0
Rectangle Color: Blue
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 19

مقصد:

ایک انٹرفیس کی وضاحت اور implement کرنے کے لئے اوا کا پروگرام لکھئیے.

پروگرام:

```
interface InterfaceExample {
  void display();
  void calculate(int x, int y);
}

class MyClass implements InterfaceExample {
  @Override
  public void display() {
    System.out.println("Method Display");
  }

  @Override
  public void calculate(int x, int y) {
    int sum = x + y;
    System.out.println("Sum:  " + sum);
  }
}

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

    MyClass obj = new MyClass();

    obj.display();

    obj.calculate(10, 5);
  }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> java InterfaceExp
Method Display
Sum:  15
```

Experiment No. 20

مقصد:

ایک قسم کے پیرامیٹر کے ساتھ generic کلاسکو ڈیمونسٹریٹ کرنے کے لئے جاوا پروگرام درج کریں.

پروگرام:

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


    GenericClass<Integer> intObj = new GenericClass<>(10);
    GenericClass<String> strObj = new GenericClass<>("Hello, World!");

    int intValue = intObj.getValue();
    System.out.println("Integer value: " + intValue);

    String strValue = strObj.getValue();
    System.out.println("String value: " + strValue);

    intObj.setValue(20);
    strObj.setValue("Hello, OpenAI!");

    int updatedIntValue = intObj.getValue();
    System.out.println("Updated integer value: " + updatedIntValue);

    String updatedStrValue = strObj.getValue();
    System.out.println("Updated string value: " + updatedStrValue);
  }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaByDanish> java ClassParamtere
Integer value: 10
String value: Hello, World!
Updated integer value: 20
Updated string value: Hello, OpenAI!
PS E:\Programming Learning\JavaByDanish>
```

Experiment No. 21

مقصد:

دو قسم کے پیرامیٹر کے ساتھ generic کلاسکو ڈیمونسٹریٹ کرنے کے لئے جاوا پروگرام درج کریں.

پروگرام:

```
class GenericClass<T, U> {
  private T value1;
  private U value2;

  public GenericClass(T value1, U value2) {
    this.value1 = value1;
    this.value2 = value2;
  }

  public T getValue1() {
    return value1;
  }

  public void setValue1(T value1) {
    this.value1 = value1;
  }

  public U getValue2() {
    return value2;
  }

  public void setValue2(U value2) {
    this.value2 = value2;
  }
}

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

    GenericClass<Integer, String> obj1 = new GenericClass<>(10, "Hello");
    GenericClass<Double, Boolean> obj2 = new GenericClass<>(3.14, true);

    int intValue = obj1.getValue1();
    String strValue = obj1.getValue2();
    System.out.println("Values of obj1: " + intValue + ", " + strValue);

    double doubleValue = obj2.getValue1();
    boolean boolValue = obj2.getValue2();
    System.out.println("Values of obj2: " + doubleValue + ", " + boolValue);
```

```
    obj1.setValue1(20);
    obj1.setValue2("World");
    obj2.setValue1(2.718);
    obj2.setValue2(false);

    intValue = obj1.getValue1();
    strValue = obj1.getValue2();
    System.out.println("Updated values of obj1: " + intValue + ", " + strValue);

    doubleValue = obj2.getValue1();

    boolValue = obj2.getValue2();
    System.out.println("Updated values of obj2: " + doubleValue + ", " + boolValue);
  }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaLab> java TwoPara
Values of obj1: 10, Hello
Values of obj2: 3.14, true
Updated values of obj1: 20, World
Updated values of obj2: 2.718, false
```

Experiment No. 22

مقصد:

Non-static nested class (nested class) کے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
public class OuterClass {
    private String outerMessage = "Hello from the outer class!";

    public void outerMethod() {
        InnerClass inner = new InnerClass();
        inner.innerMethod();
    }

    public class InnerClass {
        public void innerMethod() {
            System.out.println("Hello from the inner class!");
            System.out.println(outerMessage);
        }
    }

    public static void main(String[] args) {
        OuterClass outer = new OuterClass();
        outer.outerMethod();
    }
}
```

نتیجہ:

```
Hello from the inner class!
Hello from the outer class!
```

Experiment No. 23

مقصد:

Static nested class کے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
public class OuterClass {
    private static String message = "Hello from the outer class";

    static class NestedClass {
        public void printMessage() {
            System.out.println(message);
        }
    }

    public static void main(String[] args) {
        OuterClass.NestedClass nestedObject = new OuterClass.NestedClass();
        nestedObject.printMessage();
    }
}
```

نتیجہ:

```
Hello from the outer class
```

Experiment No. 24

مقصد:

Anonymous nested class کے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
public class OuterClass {
    public void displayMessage() {
        InnerInterface inner = new InnerInterface() {
            @Override
            public void printMessage() {
                System.out.println("Hello from the anonymous inner class");
            }
        };
        inner.printMessage();
    }

    public static void main(String[] args) {
        OuterClass outer = new OuterClass();
        outer.displayMessage();
    }

    interface InnerInterface {
        void printMessage();
    }
}
```

نتیجہ:

```
Hello from the anonymous inner class
```

Experiment No. 25

مقصد:

Local nested class کے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
public class OuterClass {
    private String message = "Hello from the outer class";

    public void displayMessage() {
        class LocalClass {
            public void printMessage() {
                System.out.println(message);
            }
        }

        LocalClass localObject = new LocalClass();
        localObject.printMessage();
    }

    public static void main(String[] args) {
        OuterClass outer = new OuterClass();
        outer.displayMessage();
    }
}
```

نتیجہ:

```
Hello from the outer class
```

Experiment No. 26

مقصد:

صارف کی وضاحت شدہ پیکیج کو وضاحت اور import کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

Example.java

```
import com.example.mypackage.MyClass;

public class Example {
    public static void main(String[] args) {
        MyClass obj = new MyClass();
        obj.displayMessage();
    }
}
```

MyClass.java

```
package com.example.mypackage;

public class MyClass {
    public void displayMessage() {
        System.out.println("Hello from MyClass!");
    }
}
```

نتیجہ:

```
javac com/example/mypackage/MyClass.java
javac Example.java
```

Then, run the program using

```
java Example
```

```
Hello from MyClass!
```

Experiment No. 27

مقصد:

Method overloadingاور overriding کے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
class Parent {
    public void display() {
        System.out.println("Parent's display method");
    }

    public void display(String message) {
        System.out.println("Parent's display method with message: " + message);
    }
}

class Child extends Parent {
    @Override
    public void display() {
        System.out.println("Child's display method");
    }

    public void display(int number) {
        System.out.println("Child's display method with number: " + number);
    }
}

public class Main {
    public static void main(String[] args) {
        Parent parent = new Parent();
        parent.display(); // Calls Parent's display method
        parent.display("Hello"); // Calls Parent's display method with message

        Child child = new Child();
        child.display(); // Calls Child's overridden display method
        child.display(42); // Calls Child's display method with number
    }
}
```

نتیجہ:

```
Parent's display method
Parent's display method with message: Hello
Child's display method
Child's display method with number: 42
```

Experiment No. 28

مقصد:

BigInteger کلاس کا استعمال کرتے ہوئےarithmetical operation پرفارم کرنے کے لئے جاوا کا پروگرام لکھیئے.

پروگرام:

```
import java.math.BigInteger;

public class BigIntegerExample {
    public static void main(String[] args) {
        BigInteger num1 = new BigInteger("123");
        BigInteger num2 = new BigInteger("45");

        // Addition
        BigInteger sum = num1.add(num2);
        System.out.println("Sum: " + sum);

        // Subtraction
        BigInteger difference = num1.subtract(num2);
        System.out.println("Difference: " + difference);

        // Multiplication
        BigInteger product = num1.multiply(num2);
        System.out.println("Product: " + product);

        // Division
        BigInteger quotient = num1.divide(num2);
        System.out.println("Quotient: " + quotient);

        // Remainder
        BigInteger remainder = num1.remainder(num2);
        System.out.println("Remainder: " + remainder);

        // Power
        BigInteger power = num1.pow(2);
        System.out.println("Power: " + power);
    }
}
```

نتیجہ:

```
Sum: 168
Difference: 78
Product: 5535
Quotient: 2
Remainder: 33
Power: 15129
```

Experiment No. 29

مقصد:

بینکنگ ٹرانزیکشن کے لئے جاوا کا پروگرام لکھئیے Balance check, Withdraw money, Deposit money :

پروگرام:

```
import java.util.Scanner;

public class BankingTransaction {
    private static double balance = 1000.00; // Initial balance

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;

        do {
            System.out.println("Banking Transactions");
            System.out.println("1. Check Balance");
            System.out.println("2. Deposit");
            System.out.println("3. Withdraw");
            System.out.println("4. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    checkBalance();
                    break;
                case 2:
                    System.out.print("Enter the amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    deposit(depositAmount);
                    break;
                case 3:
                    System.out.print("Enter the amount to withdraw: ");
                    double withdrawAmount = scanner.nextDouble();
                    withdraw(withdrawAmount);
                    break;
                case 4:
                    System.out.println("Thank you for using our banking system.
Goodbye!");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
```

```
            System.out.println();
        } while (choice != 4);
    }

    private static void checkBalance() {
        System.out.println("Current balance: $" + balance);
    }

    private static void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited $" + amount + " successfully.");
        System.out.println("Current balance: $" + balance);
    }

    private static void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient funds. Unable to withdraw $" + amount);
        } else {
            balance -= amount;
            System.out.println("Withdrawn $" + amount + " successfully.");
            System.out.println("Current balance: $" + balance);
        }
    }
}
```

نتیجہ:

```
Banking Transactions
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: 2
Enter the amount to deposit: 500
Deposited $500.0 successfully.
Current balance: $1500.0

Banking Transactions
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: 3
Enter the amount to withdraw: 2000
Insufficient funds. Unable to withdraw $2000.0

Banking Transactions
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: 3
Enter the amount to withdraw: 700
Withdrawn $700.0 successfully.
Current balance: $800.0

Banking Transactions
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: 1
Current balance: $800.0

Banking Transactions
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: 4
Thank you for using our banking system. Goodbye!
```

Experiment No. 30

مقصد:

Single thread کے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
public class SingleThreadDemo {
  public static void main(String[] args) {
    for (int i = 1; i <= 5; i++) {
      System.out.println("Main Thread: " + i);
      try {
        Thread.sleep(1000);
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    }
  }
}
```

نتیجہ:

```
PS E:\Programming Learning\JavaLab> java SingleThreadDemo
Main Thread: 1
Main Thread: 2
Main Thread: 3
Main Thread: 4
Main Thread: 5
PS E:\Programming Learning\JavaLab>
```

Experiment No. 31

مقصد:

Multi threading کے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
public class MultithreadingExample {
    public static void main(String[] args) {
        System.out.println("Starting the program.");

        Thread thread1 = new MyThread("Thread 1");
        Thread thread2 = new MyThread("Thread 2");

        thread1.start();
        thread2.start();

        System.out.println("Program finished.");
    }

    static class MyThread extends Thread {
        private String name;

        public MyThread(String name) {
            this.name = name;
        }

        @Override
        public void run() {
            for (int i = 1; i <= 5; i++) {
                System.out.println(name + ": Counter " + i);
                try {
                    Thread.sleep(500); // Pause the thread for 500 milliseconds
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

نتیجہ:

```
Starting the program.
Program finished.
Thread 1: Counter 1
Thread 2: Counter 1
Thread 1: Counter 2
Thread 2: Counter 2
Thread 1: Counter 3
Thread 2: Counter 3
Thread 1: Counter 4
Thread 2: Counter 4
Thread 1: Counter 5
Thread 2: Counter 5
```

Experiment No. 32

مقصد:

try، catchاور finally blocksکا استعمال کرتے ہوئے exception handlingکے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

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

    try {
      // Code that may throw an exception
      int result = divide(10, 0);
      System.out.println("Result: " + result);
    } catch (ArithmeticException ex) {
      // Catch the specific exception type
      System.out.println("Error: " + ex.getMessage());
    } finally {
      System.out.println("Finally block executed");
    }
  }

  public static int divide(int num1, int num2) {
    return num1 / num2;
  }
}
```

نتیجہ:

```
Error: / by zero
Finally block executed
PS E:\Programming Learning\JavaLab>
```

Experiment No. 33

مقصد:

throw اور throws کا استعمال کرتے ہوئے exception handling کے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
import java.io.*;

public class ExceptionHandlingDemo {
  public static void main(String[] args) {
    try {
      // Code that may throw an exception
      readFile("nonexistent.txt");
    } catch (FileNotFoundException ex) {
      // Catch the specific exception type
      System.out.println("Error: " + ex.getMessage());
    }
  }

  public static void readFile(String fileName) throws FileNotFoundException {
    if (!fileExists(fileName)) {
      throw new FileNotFoundException("File not found: " + fileName);
    }
    // Code to read the file goes here
    System.out.println("Reading file: " + fileName);
  }

  public static boolean fileExists(String fileName) {
    // Simulating file existence check
    return false;
  }
}
```

نتیجہ:

```
Error: File not found: nonexistent.txt
```

Experiment No. 34

مقصد:

User defined exception کے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
class MyException extends Exception {
    public MyException(String message) {
        super(message);
    }
}

public class UserDefinedExp {
    public static void main(String[] args) {
        try {
            int age = 15;
            if (age < 18) {
                throw new MyException("You must be at least 18 years old.");
            } else {
                System.out.println("Access granted! You are eligible.");
            }
        } catch (MyException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
}
```

نتیجہ:

```
Exception caught: You must be at least 18 years old.
PS E:\Programming Learning\JavaLab>
```

Experiment No. 35

مقصد:

Applet life cycle methods کے طریقہ کارکو ڈیمونسٹریٹ کرنے کے لئے جاوا کا پروگرام لکھئیے.

پروگرام:

```
import java.applet.Applet;
import java.awt.*;

public class LifeCycleApplet extends Applet {
    public void init() {
        // Initialization code
        System.out.println("Initializing the applet...");
    }

    public void start() {
        // Code to start or resume the applet
        System.out.println("Starting the applet...");
    }

    public void paint(Graphics g) {
        // Code to draw graphics on the applet
        g.drawString("Applet Life Cycle", 50, 50);
    }

    public void stop() {
        // Code to stop or suspend the applet
        System.out.println("Stopping the applet...");
    }

    public void destroy() {
        // Code to clean up resources used by the applet
        System.out.println("Destroying the applet...");
    }
}
```

نتیجہ:

```
javac LifeCycleApplet.java

appletviewer LifeCycleApplet.java
```

Experiment No. 36

مقصد:

بٹن کلک کے ساتھ بیک -گراؤنڈ کے رنگ کو تبدیل کرنے کا پروگرام ڈیمونسٹریٹ کرنے کے لئے ActionListener کی مدد سے اس ایونٹ کوhandle کریں.

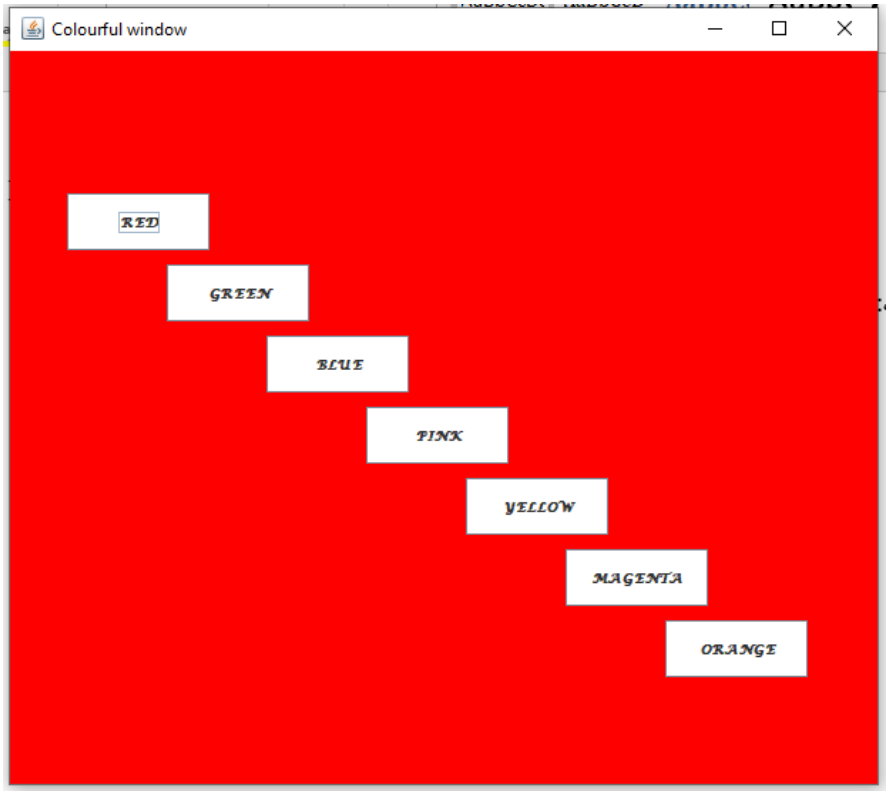
پروگرام:

```
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
public class ChangeBackground extends JFrame implements ActionListener {

    JFrame frame;
    JButton red, green, blue, pink, yellow, magenta, orange;

    ChangeBackground() {

        Font font = new Font("Lucida Calligraphy", Font.BOLD, 9);
        frame = new JFrame("Colourful window");
        Color c = new Color(255, 255, 255);

        red = new JButton("RED");
        red.setBounds(40, 100, 100, 40);
        red.setFont(font);
        red.setBackground(c);

        green = new JButton("GREEN");
        green.setBounds(110, 150, 100, 40);
        green.setFont(font);
        green.setBackground(c);

        blue = new JButton("BLUE");
        blue.setBounds(180, 200, 100, 40);
        blue.setFont(font);
        blue.setBackground(c);

        pink = new JButton("PINK");
        pink.setBounds(250, 250, 100, 40);
        pink.setFont(font);
        pink.setBackground(c);

        yellow = new JButton("YELLOW");
        yellow.setBounds(320, 300, 100, 40);
        yellow.setFont(font);
```

```
        yellow.setBackground(c);

        magenta = new JButton("MAGENTA");
        magenta.setBounds(390, 350, 100, 40);
        magenta.setFont(font);
        magenta.setBackground(c);

        orange = new JButton("ORANGE");
        orange.setBounds(460, 400, 100, 40);
        orange.setFont(font);
        orange.setBackground(c);

        frame.add(red);
        frame.add(green);
        frame.add(blue);
        frame.add(pink);
        frame.add(yellow);
        frame.add(magenta);
        frame.add(orange);

        red.addActionListener(this);
        green.addActionListener(this);
        blue.addActionListener(this);
        pink.addActionListener(this);
        yellow.addActionListener(this);
        magenta.addActionListener(this);
        orange.addActionListener(this);

        frame.getContentPane().setBackground(Color.black);
        frame.setLayout(null);
        frame.setSize(650, 600);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public void actionPerformed(ActionEvent ae)
    {

        if(ae.getSource() == red)
            frame.getContentPane().setBackground(Color.RED);
        if(ae.getSource() == green)
            frame.getContentPane().setBackground(Color.GREEN);
        if(ae.getSource() == blue)
            frame.getContentPane().setBackground(Color.BLUE);
        if(ae.getSource() == pink)
            frame.getContentPane().setBackground(Color.PINK);
```

```
        if(ae.getSource() == yellow)
            frame.getContentPane().setBackground(Color.YELLOW);
        if(ae.getSource() == magenta)
            frame.getContentPane().setBackground(Color.MAGENTA);
        if(ae.getSource() == orange)
            frame.getContentPane().setBackground(Color.ORANGE);

    }

    public static void main(String[] args) {

        new ChangeBackground();

    }

}
```

نتیجہ:



Experiment No. 37

مقصد:

ساکٹ پروگرامنگ کا استعمال کرتے ہوئے کلائنٹ سرور ایپلیکیشن ڈیولپ کریں.

پروگرام:

```
//MyServer.java
import java.io.*;
import java.net.*;

public class MyServer {
  public static void main(String[] args) {
    try {
      ServerSocket ss = new ServerSocket(6666);
      Socket s = ss.accept();// establishes connection
      DataInputStream dis = new DataInputStream(s.getInputStream());
      String str = (String) dis.readUTF();
      System.out.println("message= " + str);
      ss.close();
    } catch (Exception e) {
      System.out.println(e);
    }
  }
```

```
//MyClient.java
import java.io.*;
import java.net.*;

public class MyClient {
  public static void main(String[] args) {
    try {
      Socket s = new Socket("localhost", 6666);
      DataOutputStream dout = new DataOutputStream(s.getOutputStream());
      dout.writeUTF("Hello Server");
      s.close();
    } catch (Exception e) {
      System.out.println(e);
    }
  }
}
```

نتیجہ:

When i execute MyServer then blinking cursor

```
PS E:\Programming Learning\JavaLab> java MyServer
```

When i execute MyClient then Client Send msg to server

```
PS E:\Programming Learning\JavaLab> javac MyClient.java
PS E:\Programming Learning\JavaLab> java MyClient
PS E:\Programming Learning\JavaLab>
```

MyServe received msg Hello Server

```
PS E:\Programming Learning\JavaLab> javac MyServer.java
PS E:\Programming Learning\JavaLab> java MyServer
message= Hello Server
PS E:\Programming Learning\JavaLab>
```

Experiment No. 38

مقصد:

ریاضی عملیات(arithmetic operation) انجام دینے کے لئے جاوا فریم پروگرام لکھیں.

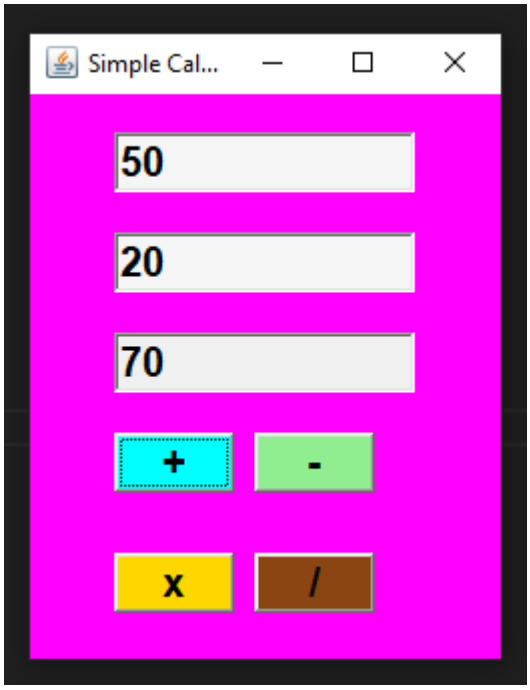
پروگرام:

```
import java.awt.*;
import java.awt.event.*;

public class SimpleCalculator extends Frame implements ActionListener {
  TextField textField1, textField2, textField3;
  Button btn1, btn2, btn3, btn4;
  Label label1, labe2, label3;

  SimpleCalculator() {

    setTitle("Simple Calculator");

    Color c1 = new Color(245,245,245);
    Color c2 = new Color(0,255,255);
    Color c3 = new Color   (255,215,0);
    Color c4 = new Color(144, 238, 144);
    Color c5 = new Color(139,69,19);

    textField1 = new TextField();
    textField1.setFont(new Font("Raleway", Font.BOLD, 20));
    textField1.setBounds(50, 50, 150, 30);
    textField1.setBackground(c1);

    textField2 = new TextField();
    textField2.setFont(new Font("Raleway", Font.BOLD, 20));
    textField2.setBounds(50, 100, 150, 30);
    textField2.setBackground(c1);

    textField3 = new TextField();
    textField3.setFont(new Font("Raleway", Font.BOLD, 20));
    textField3.setBounds(50, 150, 150, 30);
    textField3.setEditable(false);

    btn1 = new Button("+");
    btn1.setFont(new Font("Raleway", Font.BOLD, 20));
    btn1.setBounds(50, 200, 60, 30);
    btn1.setBackground(c2);

    btn2 = new Button("-");
    btn2.setFont(new Font("Raleway", Font.BOLD, 20));
    btn2.setBounds(120, 200, 60, 30);
```

```
  btn2.setBackground(c4);

  btn3 = new Button("x");
  btn3.setFont(new Font("Raleway", Font.BOLD, 20));
  btn3.setBounds(50, 260, 60, 30);
  btn3.setBackground(c3);

  btn4 = new Button("/");
  btn4.setFont(new Font("Raleway", Font.BOLD, 20));
  btn4.setBounds(120, 260, 60, 30);
  btn4.setBackground(c5);

  btn1.addActionListener(this);
  btn2.addActionListener(this);
  btn3.addActionListener(this);
  btn4.addActionListener(this);

  add(textField1);
  add(textField2);
  add(textField3);
  add(btn1);
  add(btn2);
  add(btn3);
  add(btn4);

  setSize(300, 300);
  setLayout(null);
  setVisible(true);

  Color bg = new Color(255,0,255);
  setBackground(bg);

  addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
      dispose();
    }
  });

}

public void actionPerformed(ActionEvent e) {
  String s1 = textField1.getText();
  String s2 = textField2.getText();

  int a = Integer.parseInt(s1);
  int b = Integer.parseInt(s2);
  int c = 0;
```

```
    if (e.getSource() == btn1) {
      c = a + b;
    } else if (e.getSource() == btn2) {
      c = a - b;
    }
    else if (e.getSource() == btn3) {
      c = a * b;
    }
    else if (e.getSource() == btn4) {
      c = a / b;
    }
    String result = String.valueOf(c);
    textField3.setText(result);
  }

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

نتیجہ:



Experiment No. 39

مقصد:

(CGPA Calculator) انجام دینے کے لئے جاوا فریم پروگرام لکھیں.

پروگرام:

```
import java.util.Scanner;

public class CgpaCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("---------- CGPA Calculator ----------");
        System.out.println("-------------------------------------");

        // Get the number of courses
        System.out.print("Enter the number of courses: ");
        int numCourses = scanner.nextInt();

        double totalGradePoints = 0.0;
        int totalCreditHours = 0;

        // Get the grades and credit hours for each course
        for (int i = 1; i <= numCourses; i++) {
            System.out.println("\nCourse " + i + ":");
            System.out.print("Enter the grade points 0-10: ");
            double gradePoints = scanner.nextDouble();

            System.out.print("Enter the credit hours: ");
            int creditHours = scanner.nextInt();

            totalGradePoints += gradePoints * creditHours;
            totalCreditHours += creditHours;
        }

        // Calculate CGPA
        double cgpa = totalGradePoints / totalCreditHours;
        double percentage = cgpa * 9.5;

        // Display the CGPA
        System.out.println("\nCGPA: " + cgpa);
        System.out.println("\nPercentage: " + percentage + " %");

        scanner.close();
    }
}
```

نتیجہ:

```
PS E:\Programming Learning\Java Learning> java CgpaCalculator
---------- CGPA Calculator ----------
-------------------------------------
Enter the number of courses: 3

Course 1:
Enter the grade points 0-10: 55
Enter the credit hours: 70

Course 2:
Enter the grade points 0-10: 65
Enter the credit hours: 65

Course 3:
Enter the grade points 0-10: 50
Enter the credit hours: 80
```

Experiment No. 40

مقصد:

Electricity Bill انجام دینے کے لئے جاوا فریم پروگرام لکھیں.

پروگرام:

```
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;

public class ElectricityBill {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("------------ Electricity Bill Generator --------------");
        System.out.println("-----------------------------------------------------");

        // Get user input for customer details
        System.out.print("Enter customer name: ");
        String customerName = scanner.nextLine();

        System.out.print("Enter customer ID: ");
        String customerID = scanner.nextLine();

        System.out.print("Enter units consumed: ");
        int unitsConsumed = scanner.nextInt();

        // Get current date and time
        LocalDateTime currentDateTime = LocalDateTime.now();
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy
HH:mm:ss");
        String formattedDateTime = currentDateTime.format(formatter);

        // Calculate bill amount
        double billAmount = calculateBillAmount(unitsConsumed);

        // Generate bill
        System.out.println("\n------------------ Electricity Bill ------------------
");
        System.out.println("Date and Time: " + formattedDateTime);
        System.out.println("-----------------------------------------------------");
        System.out.println("Customer Name: " + customerName);
        System.out.println("Customer ID: " + customerID);
        System.out.println("-----------------------------------------------------");
        System.out.printf("%-20s %-20s %-10s%n", "Units Consumed", "Rate (Rs./Unit)",
"Amount (Rs.)");
```

```
System.out.println("-----------------------------------------------------------");
System.out.printf("%-20d %-20.2f %-10.2f%n", unitsConsumed, getRatePerUnit(),
billAmount);
        System.out.println("-----------------------------------------------------------");

        scanner.close();
    }

    private static double calculateBillAmount(int unitsConsumed) {
        // Calculation logic for bill amount, unit
        double perUnitRate = getRatePerUnit();
        double billAmount = perUnitRate * unitsConsumed;
        return billAmount;
    }

    private static double getRatePerUnit() {
        return 7.50; // Sample rate per unit
    }
}
```

نتیجہ:

```
----------- Electricity Bill Generator -------------
------------------------------------------------------
Enter customer name: Mohtesham Pasha
Enter customer ID: 5636
Enter units consumed: 350

------------------ Electricity Bill ------------------
Date and Time: 22-06-2023 10:27:24
------------------------------------------------------
Customer Name: Mohtesham Pasha
Customer ID: 5636
------------------------------------------------------
Units Consumed       Rate (Rs./Unit)      Amount (Rs.)
------------------------------------------------------
350                  7.50                 2625.00
------------------------------------------------------
PS E:\Programming Learning\Java Learning>
```

Experiment No. 41

مقصد:

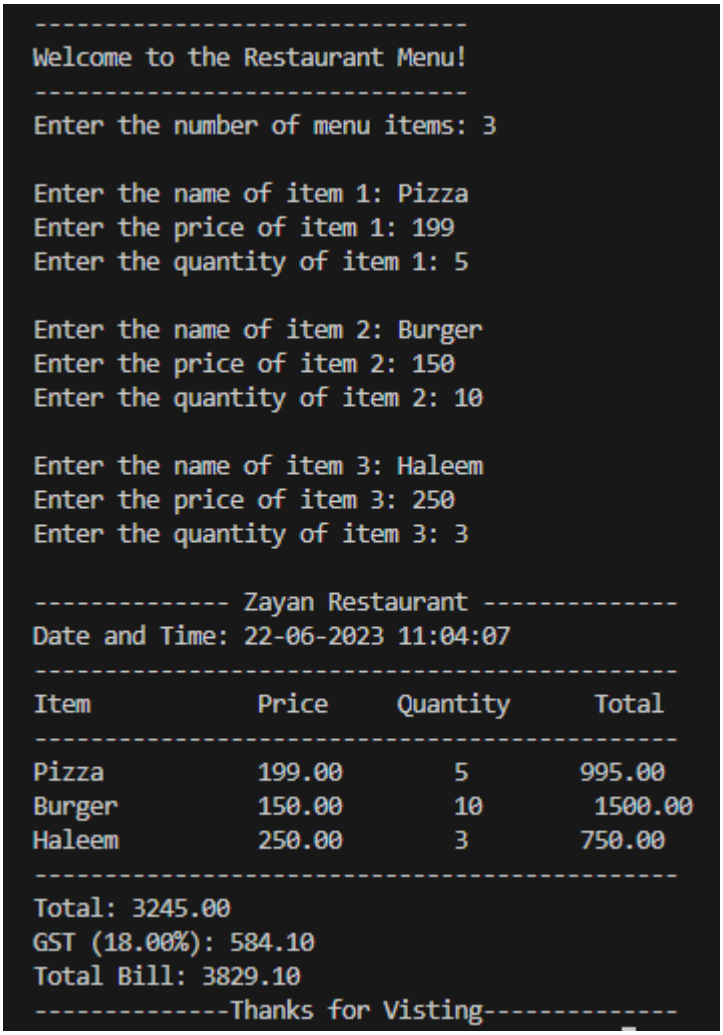
Restaurant Menu انجام دینے کے لئے جاوا فریم پروگرام لکھیں.

پروگرام:

```
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;

public class RestaurantMenu {
    private static final double GST_RATE = 0.18;

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("--------------------------------");
        System.out.println("Welcome to the Restaurant Menu!");
        System.out.println("--------------------------------");

        // Get user input for menu size
        System.out.print("Enter the number of menu items: ");
        int numItems = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character

        // Initialize arrays to store menu items, prices, and quantities
        String[] menuItems = new String[numItems];
        double[] itemPrices = new double[numItems];
        int[] itemQuantities = new int[numItems];

        // Get user inputs for item details
        for (int i = 0; i < numItems; i++) {
            System.out.print("\nEnter the name of item " + (i + 1) + ": ");
            menuItems[i] = scanner.nextLine();

            System.out.print("Enter the price of item " + (i + 1) + ": ");
            itemPrices[i] = scanner.nextDouble();

            System.out.print("Enter the quantity of item " + (i + 1) + ": ");
            itemQuantities[i] = scanner.nextInt();

            scanner.nextLine(); // Consume the newline character
        }
```

```
        // Calculate total price
        double totalPrice = 0.0;
        for (int i = 0; i < numItems; i++) {
            totalPrice += itemPrices[i] * itemQuantities[i];
        }

        // Calculate GST amount
        double gstAmount = totalPrice * GST_RATE;

        // Calculate total bill including GST
        double totalBill = totalPrice + gstAmount;

        // Get current date and time for bill
        LocalDateTime currentDateTime = LocalDateTime.now();
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy
HH:mm:ss");
        String formattedDateTime = currentDateTime.format(formatter);

        // Generate bill for zayan restaurant
        System.out.println("\n-------------- Zayan Restaurant --------------");
        System.out.println("Date and Time: " + formattedDateTime);
        System.out.println("------------------------------------------------");
        System.out.println("Item            Price     Quantity      Total");
        System.out.println("------------------------------------------------");

        for (int i = 0; i < numItems; i++) {
            double itemTotal = itemPrices[i] * itemQuantities[i];
            System.out.printf("%-15s %.2f        %d        %.2f%n",
                    menuItems[i], itemPrices[i], itemQuantities[i], itemTotal);
        }

        System.out.println("------------------------------------------------");
        System.out.printf("Total: %.2f%n", totalPrice);
        System.out.printf("GST (%.2f%%): %.2f%n", (GST_RATE * 100), gstAmount);
        System.out.printf("Total Bill: %.2f%n", totalBill);
        System.out.println("--------------Thanks for Visting--------------");

        scanner.close();
    }
}
```

نتیجہ:



```
------------------------------
Welcome to the Restaurant Menu!
------------------------------
Enter the number of menu items: 3

Enter the name of item 1: Pizza
Enter the price of item 1: 199
Enter the quantity of item 1: 5

Enter the name of item 2: Burger
Enter the price of item 2: 150
Enter the quantity of item 2: 10

Enter the name of item 3: Haleem
Enter the price of item 3: 250
Enter the quantity of item 3: 3

-------------- Zayan Restaurant --------------
Date and Time: 22-06-2023 11:04:07
------------------------------------------------
Item            Price     Quantity      Total
------------------------------------------------
Pizza           199.00        5        995.00
Burger          150.00        10        1500.00
Haleem          250.00        3        750.00
------------------------------------------------
Total: 3245.00
GST (18.00%): 584.10
Total Bill: 3829.10
--------------Thanks for Visting--------------
```