**Assignment 1(A): Write java program using java compiler on DOS prompt to print Hello World**

public class q1\_a {

public static void main(String[] args) throws Exception {

System.out.println("Hello, World!");

}

}

**Assignment 1(B): Write java program using java compiler on DOS prompt to addition of two numbers.**

public class q1\_b {

public static void main(String[] args) {

int a = 10;

int b = 20;

int c = a + b;

System.out.println("Sum of a and b is: " + c);

}

}

**Assignment 2(A): Write a java program to find the Fibonacci series**

import java.util.Scanner;

public class q2\_a {

    public static void main(String[] args) {

        int t1 = 0, t2 = 1, next\_term = t1 + t2;

        int n;

        System.out.println("Enter the number of terms: ");

        Scanner sc = new Scanner(System.in);

        n = sc.nextInt();

        sc.close(); // Close the Scanner object

        System.out.println(String.format("Fibonacci Series till %s terms:", n));

        for (int i = 1; i <= n; ++i) {

            System.out.print(t1 + ", ");

            next\_term = t1 + t2;

            t1 = t2;

            t2 = next\_term;

        }

    }

}

**Assignment 2(B): Write a Program in java to change Upper case to Lower case & Vice Versa**

import java.util.Scanner;

public class q2\_b {

    static String str;

    public static void main(String[] args) {

        System.out.println("Enter the String :");

        Scanner sc = new Scanner(System.in);

        str = sc.nextLine();

        sc.close();

        // Convert string to uppercase

        String uppercaseStr = str.toUpperCase();

        System.out.println("Uppercase String: " + uppercaseStr);

        // Convert string to lowercase

        String lowercaseStr = str.toLowerCase();

        System.out.println("Lowercase String: " + lowercaseStr);

    }

}

**Assignment 3: Write a java program to display the employee details using Scanner class**

import java.util.Scanner;

public class q3 {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        System.out.print("Enter employee name: ");

        String name = obj.nextLine();

        System.out.print("Enter employee ID: ");

        int id = obj.nextInt();

        System.out.print("Enter employee salary: ");

        int salary = obj.nextInt();

        System.out.println("Employee details:");

        System.out.println("Name: " + name);

        System.out.println("ID: " + id);

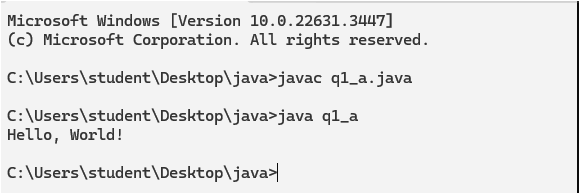
        System.out.println("Salary: " + salary);

        obj.close();

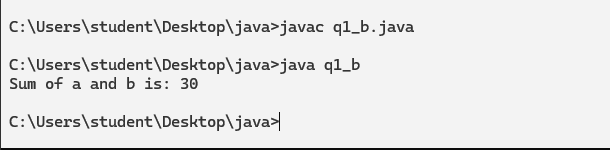
    }

}

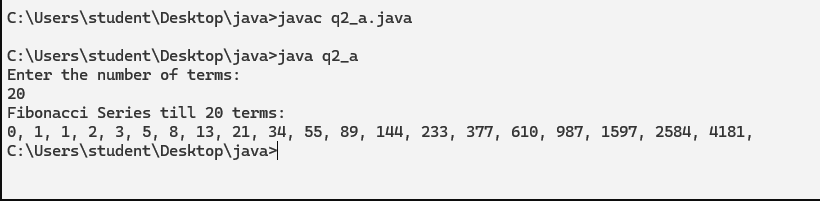
**Assignment 1(A): Output**



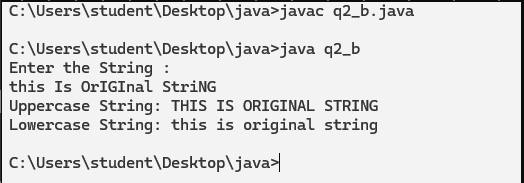
**Assignment 1(B): Output**



**Assignment 2(A): Output**



**Assignment 2(B): Output**



**Assignment 3: Output**



**Assignment 4: Method Overloading**

class MyClass {

public void print(int num) {

System.out.println("Printing number: " + num);

}

public void print(String str) {

System.out.println("Printing string: " + str);

}

public void print(int num1, int num2) {

System.out.println("Printing numbers: " + num1 + " and " + num2);

}

}

public class MethodOverloading {

public static void main(String[] args) {

MyClass myObj = new MyClass();

myObj.print(5);

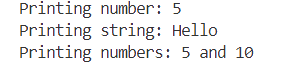
myObj.print("Hello");

myObj.print(5, 10);

}

}

**Assignment 4: Method Overloading (Output)**



**Assignment 4: Method Overriding**

class Animal {

    public void makeSound() {

        System.out.println("The animal makes a sound");

    }

}

class Dog extends Animal {

    @Override

    public void makeSound() {

        System.out.println("The dog barks");

    }

}

public class Methodoverriding {

    public static void main(String[] args) {

        Animal animal = new Animal();

        animal.makeSound(); // Output: The animal makes a sound

        Dog dog = new Dog();

        dog.makeSound(); // Output: The dog barks

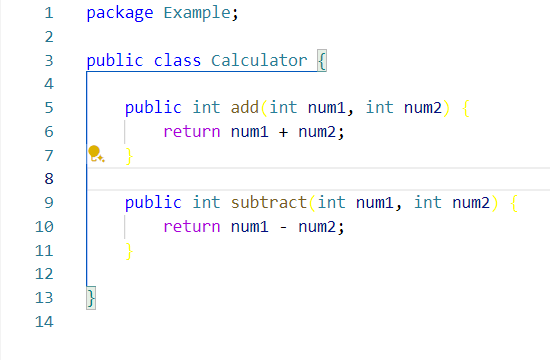
    }

}

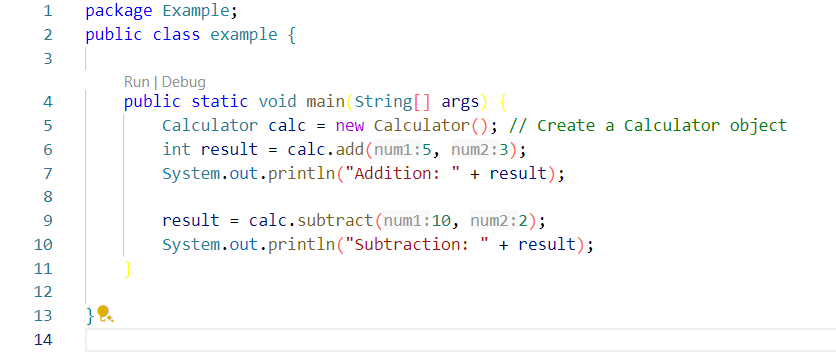
**Assignment 4: Method Overriding (Output)**



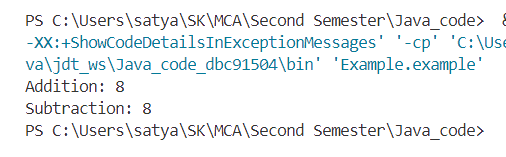
**Assignment 6:Create User Defined package**

Calculator.java

example.java



**Assignment 6: Output**



**Assignment 5: Implementation of multi-thread application**

public class MultiThreadExample {

    public static void main(String[] args) {

        Thread thread1 = new Thread(new MyRunnable("Thread 1"));

        Thread thread2 = new Thread(new MyRunnable("Thread 2"));

        Thread thread3 = new Thread(new MyRunnable("Thread 3"));

        thread1.start();

        thread2.start();

        thread3.start();

    }

    static class MyRunnable implements Runnable {

        private String name;

        public MyRunnable(String name) {

            this.name = name;

        }

        @Override

        public void run() {

            System.out.println("Thread " + name + " is running");

            for (int i = 1; i <= 10; i++) {

                System.out.println("Thread " + name + " prints " + i);

                try {

                    Thread.sleep(1000); // Sleep for 1 second

                } catch (InterruptedException e) {

                    e.printStackTrace();

                }

            }

        }

    }

}

**Assignment 5: OUTPUT**

Thread Thread 1 is running

Thread Thread 2 is running

Thread Thread 3 is running

Thread Thread 3 prints 1

Thread Thread 2 prints 1

Thread Thread 1 prints 1

Thread Thread 3 prints 2

Thread Thread 1 prints 2

Thread Thread 2 prints 2

Thread Thread 2 prints 3

Thread Thread 3 prints 3

Thread Thread 1 prints 3

Thread Thread 3 prints 4

Thread Thread 2 prints 4

Thread Thread 1 prints 4

Thread Thread 2 prints 5

Thread Thread 1 prints 5

Thread Thread 3 prints 5

Thread Thread 2 prints 6

Thread Thread 3 prints 6

Thread Thread 1 prints 6

Thread Thread 2 prints 7

Thread Thread 3 prints 7

Thread Thread 1 prints 7

Thread Thread 2 prints 8

Thread Thread 3 prints 8

Thread Thread 1 prints 8

Thread Thread 3 prints 9

Thread Thread 1 prints 9

Thread Thread 2 prints 9

Thread Thread 1 prints 10

Thread Thread 3 prints 10

Thread Thread 2 prints 10

**Assignment 7: Program to demonstrate exception handling**

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 e) {

            // Exception handling code

            System.out.println("Error: Division by zero");

        }

    }

    public static int divide(int num1, int num2) {

        return num1 / num2;

    }

}

**Assignment 7: OUTPUT**

Error: Division by zero

**Assignment 8: Applet program that displays a simple message**

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import javax.swing.JComponent;

import javax.swing.JFrame;

public class MyApplet extends JComponent {

    public void paintComponent(Graphics g) {

        super.paintComponent(g);

        // Set the color

        g.setColor(Color.BLUE);

        // Set the font style and size

        Font font = new Font("Arial", Font.BOLD, 24);

        g.setFont(font);

        // Draw the text

        g.drawString("Hello, World!", 20, 20);

    }

    public static void main(String[] args) {

        MyAppletFrame frame = new MyAppletFrame();

        frame.setVisible(true);

    }

}

class MyAppletFrame extends JFrame {

    public MyAppletFrame() {

        setTitle("My Applet");

        setSize(600, 350);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        // Add the MyApplet component to the frame

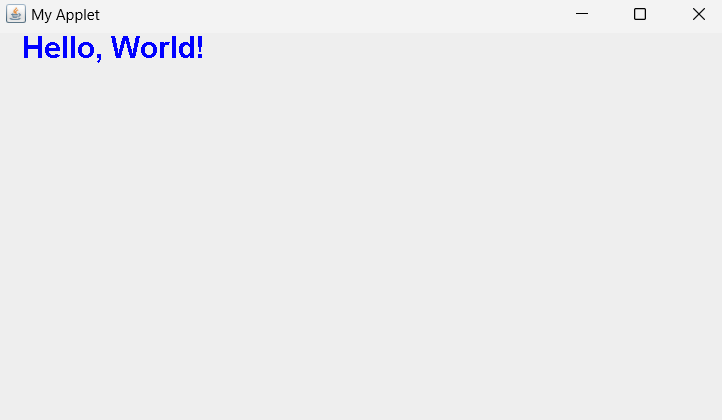
        MyApplet applet = new MyApplet();

        add(applet);

    }

}

**OUTPUT :**



**Assignment 9: Program that works as a simple calculator**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Calculator2GUI {

    private JFrame frame;

    private JTextField num1Field;

    private JTextField num2Field;

    private JComboBox<String> operationBox;

    private JButton calculateButton;

    private JLabel resultLabel;

    public Calculator2GUI() {

        frame = new JFrame("Calculator");

        frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        frame.setSize(600, 300);

        num1Field = new JTextField();

        num2Field = new JTextField();

        String[] operations = { "Addition", "Subtraction", "Multiplication", "Division" };

        operationBox = new JComboBox<>(operations);

        calculateButton = new JButton("Calculate");

        calculateButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                double num1 = Double.parseDouble(num1Field.getText());

                double num2 = Double.parseDouble(num2Field.getText());

                String operation = (String) operationBox.getSelectedItem();

                double result = 0;

                switch (operation) {

                    case "Addition":

                        result = num1 + num2;

                        break;

                    case "Subtraction":

                        result = num1 - num2;

                        break;

                    case "Multiplication":

                        result = num1 \* num2;

                        break;

                    case "Division":

                        result = num1 / num2;

                        break;

                }

                resultLabel.setText("Result: " + result);

            }

        });

        resultLabel = new JLabel("Result: ");

        JPanel panel = new JPanel();

        panel.setLayout(new GridLayout(5, 2));

        panel.add(new JLabel("Number 1: "));

        panel.add(num1Field);

        panel.add(new JLabel("Number 2: "));

        panel.add(num2Field);

        panel.add(new JLabel("Operation: "));

        panel.add(operationBox);

        panel.add(calculateButton);

        panel.add(resultLabel);

        frame.add(panel);

        frame.setVisible(true);

    }

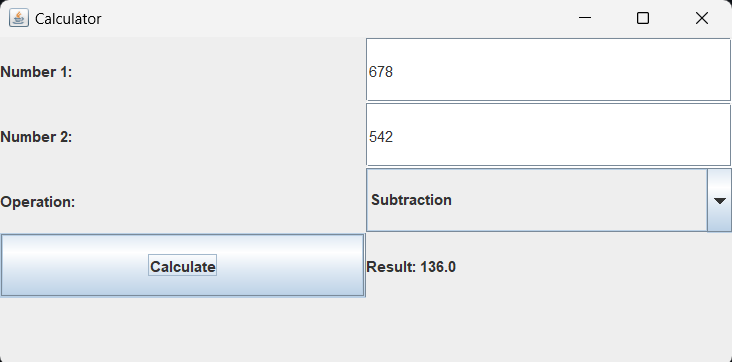
    public static void main(String[] args) {

        new Calculator2GUI();

    }

}

**Assignment 9: OUTPUT**



**Assignment 10: Java program for handling Mouse events and Key events**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class EventHandling extends JFrame implements MouseListener, KeyListener {

private JLabel mouseEventLabel;

private JLabel keyEventLabel;

public EventHandling() {

setTitle("Mouse and Key Event Handling Example");

setSize(500, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

setLayout(new GridLayout(2, 1));

mouseEventLabel = new JLabel("Mouse Events: ", SwingConstants.CENTER);

mouseEventLabel.setFont(new Font("Arial", Font.BOLD, 16));

add(mouseEventLabel);

keyEventLabel = new JLabel("Key Events: ", SwingConstants.CENTER);

keyEventLabel.setFont(new Font("Arial", Font.BOLD, 16));

add(keyEventLabel);

addMouseListener(this);

addKeyListener(this);

setFocusable(true);

setVisible(true);

}

@Override

public void mouseClicked(MouseEvent e) {

mouseEventLabel.setText("Mouse Clicked at (" + e.getX() + ", " + e.getY() + ")");

}

@Override

public void mousePressed(MouseEvent e) {

mouseEventLabel.setText("Mouse Pressed at (" + e.getX() + ", " + e.getY() + ")");

}

@Override

public void mouseReleased(MouseEvent e) {

mouseEventLabel.setText("Mouse Released at (" + e.getX() + ", " + e.getY() + ")");

}

@Override

public void mouseEntered(MouseEvent e) {

mouseEventLabel.setText("Mouse Entered the window");

}

@Override

public void mouseExited(MouseEvent e) {

mouseEventLabel.setText("Mouse Exited the window");

}

@Override

public void keyTyped(KeyEvent e) {

keyEventLabel.setText("Key Typed: " + e.getKeyChar());

}

@Override

public void keyPressed(KeyEvent e) {

keyEventLabel.setText("Key Pressed: " + e.getKeyChar());

}

@Override

public void keyReleased(KeyEvent e) {

keyEventLabel.setText("Key Released: " + e.getKeyChar());

}

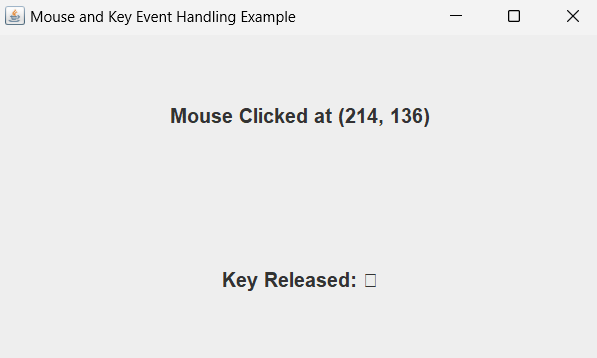
public static void main(String[] args) {

SwingUtilities.invokeLater(EventHandling::new);

}

}

**OUTPUT :**



**Assignment 11: Java program for handling Key events**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class KeyEventHandling extends JFrame implements KeyListener {

private JLabel keyEventLabel;

public KeyEventHandling() {

setTitle("Key Event Handling ");

setSize(500, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

setLayout(new BorderLayout());

keyEventLabel = new JLabel("Press any key", SwingConstants.CENTER);

keyEventLabel.setFont(new Font("Arial", Font.BOLD, 16));

add(keyEventLabel, BorderLayout.CENTER);

addKeyListener(this);

setFocusable(true);

setVisible(true);

}

@Override

public void keyTyped(KeyEvent e) {

keyEventLabel.setText("Key Typed: " + e.getKeyChar());

}

@Override

public void keyPressed(KeyEvent e) {

keyEventLabel.setText("Key Pressed: " + KeyEvent.getKeyText(e.getKeyCode()));

}

@Override

public void keyReleased(KeyEvent e) {

keyEventLabel.setText("Key Released: " + KeyEvent.getKeyText(e.getKeyCode()));

}

public static void main(String[] args) {

SwingUtilities.invokeLater(KeyEventHandling::new);

}

}

**OUTPUT :**

