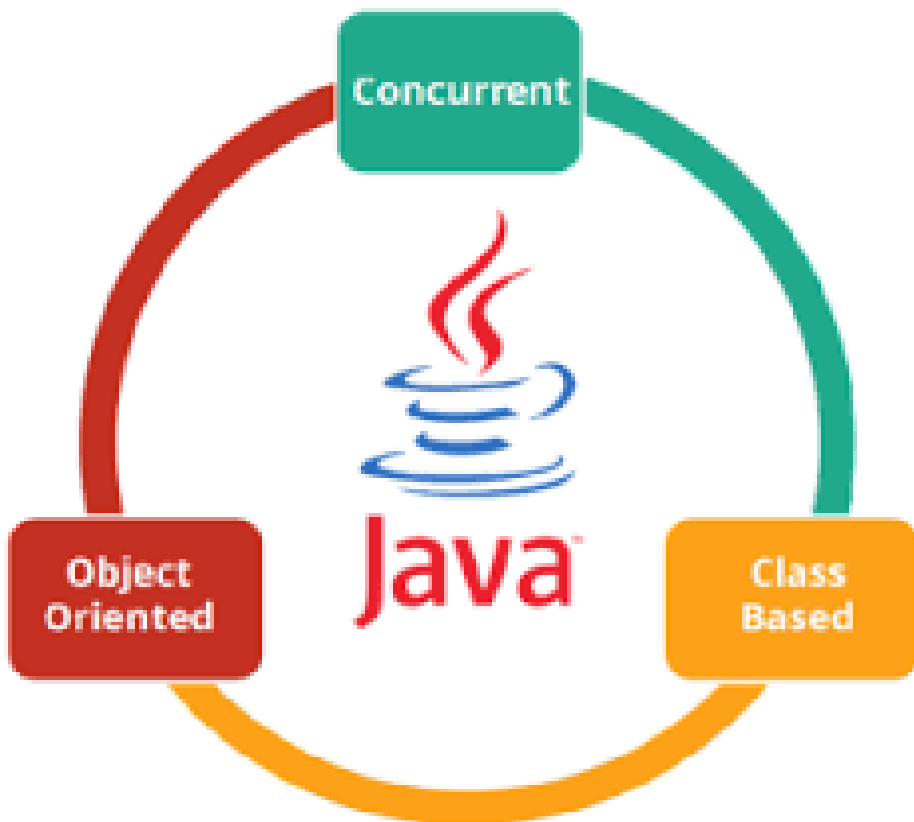
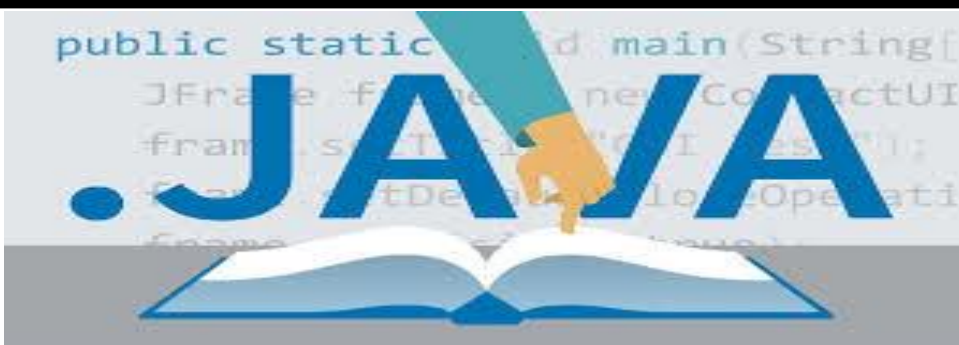




2019

PROBLEMS AND SOLUTIONS



UMM AL QURA
UNIVERSITY

Ali Al Najjar

2/11/2019

Sales Tax Problem Code: P1

Write a program that will ask the user to enter the amount of a purchase. The program should then compute the state and county sales tax. Assume the state sales tax is 4 percent and the county sales tax is 2 percent. The program should display the amount of the purchase, the state sales tax, the county sales tax, the total sales tax, and the total of the sale (which is the sum of the amount of purchase plus the total sales tax).

Sample Input and Output expected.

Enter the purchase amount: 2000

Purchase amount: \$2000.0

State tax: \$80.0

County tax: \$40.0

Total tax: \$120.0

Total cost: \$2120.0

Solution: Sales Tax Problem Code: P1

```
/*
   This program demonstrates a solution to the
   Sales Tax
 */
package saletax;

import java.util.Scanner; // Needed for the Scanner
class
public class SalesTax {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Constants
        final double STATE_RATE = 0.04; // State tax
rate
```

```

    final double COUNTY_RATE = 0.02; // County tax
rate

    // Variables
    double purchase;           // Amount of purchase
    double stateTax;           // Amount of state sales
tax
    double countyTax;          // Amount of county
sales tax
    double totalTax;           // Total sales tax
    double totalCost;          // Total cost of the
purchase

    // Create a Scanner object for keyboard input.
    Scanner keyboard = new Scanner(System.in);

    // Get the amount of the purchase.
    System.out.print("Enter the purchase amount: ");
    purchase = keyboard.nextDouble();

    // Calculate the state sales tax.
    stateTax = purchase * STATE_RATE;

    // Calculate county sales tax.
    countyTax = purchase * COUNTY_RATE;

    // Calculate the total sales tax.
    totalTax = stateTax + countyTax;

    // Calculate the total purchase cost.
    totalCost = purchase + totalTax;

    // Display the results.
    System.out.println("Purchase amount: $" +
purchase);
    System.out.println("State tax: $" + stateTax);
    System.out.println("County tax: $" + countyTax);
    System.out.println("Total tax: $" + totalTax);
    System.out.println("Total cost: $" + totalCost);
}}

```

Random Number Guessing Game Problem Code: P2

Write a program that generates a random number and asks the user to guess what the number is. If the user's guess is higher than the random number, the program should display "Too high, try again." If the user's guess is lower than the random number, the program should display "Too low, try again." The program should use a loop that repeats until the user correctly guesses the random number.

Input and output expected

run:

I'm thinking of a number.

Guess what it is: 11

Sorry, that's too high.

Guess again: 9

Sorry, that's too high.

Guess again: 8

Congratulations! You guessed it!

I was thinking of the number 8.

BUILD SUCCESSFUL (total time: 25 seconds)

SOLUTION: Random Number Guessing Game Problem Code: 2

```
/**
    This program demonstrates a solution to the
    Random Number Guessing Game programming challenge.
 */
package random.number.guessing.game;
import java.util.Random;
import java.util.Scanner;
public class RandomNumberGuessingGame {

    /**
     * @param args the command line arguments
    */
}
```

```

    */
    public static void main(String[] args) {

    {
        // Constant for the maximum random number
        final int MAX_NUMBER = 20;

        // Variables
        int guess;        // To hold the user's guess
        int randNum;      // To hold a random number

        // Create a Scanner object for keyboard input.
        Scanner keyboard = new Scanner(System.in);

        // Create a Random object.
        Random rand = new Random();

        // Generate a random number.
        randNum = rand.nextInt(MAX_NUMBER);

        // Get the user's guess.
        System.out.println("I'm thinking of a number.");
        System.out.print("Guess what it is: ");
        guess = keyboard.nextInt();

        // Respond to the user's guess.
        while (guess != randNum)
        {
            if (guess < randNum)
            {
                System.out.println("No, that's too low.");
            }
            else if (guess > randNum)
            {
                System.out.println("Sorry, that's too high.");
            }

            // Get another guess.
            System.out.print("Guess again: ");
            guess = keyboard.nextInt();
        }
    }
}

```

```

    }

    // Congratulate the user.
    System.out.println("Congratulations! You guessed
it!");
    System.out.println("I was thinking of the number " +
        randNum + ".");
}
}
}

```

isPrime Method: Problem Code: P3

A prime number is a number that is evenly divisible only by itself and 1. For example, the number 5 is prime because it can be evenly divided only by 1 and 5. The number 6, however, is not prime because it can be divided evenly by 1, 2, 3, and 6.

Write a method named `isPrime`, which takes an integer as an argument and returns `true` if the argument is a prime number, or `false` otherwise.

Demonstrate the method in a complete program.

Tip: Recall that the `%` operator divides one number by another and returns the remainder of the division. In an expression such as `num1 % num2`, the `%` operator will return 0 if `num1` is evenly divisible by `num2`.

Input and output Expected:

Enter a number: 13

That is a prime Number.

Solution: isPrime Method: Problem Code: P3

```

/*
    This program demonstrates a solution to the
    isPrime Method programming challenge.
*/
package isprimemethod;

import javax.swing.JOptionPane;
public class IsPrimeMethod {
    /**
        The isPrime method determines whether a number is
        prime.
    */
}

```

```

        @param num The number to check.
        @return true if the number is prime, false
otherwise.
    */
    public static boolean isPrime(int num)
    {
        boolean divisorFound = false;
        int div = 2;

        while(div < num && !divisorFound)
        {
            if ((num % div) == 0)
                divisorFound = true;
            div++;
        }

        return !divisorFound;
    }

    public static void main(String[] args) {
        String input;           // To hold keyboard input
        String message;         // Message to display
        int number;              // Number to check for prime

        // Get the number.
        input = JOptionPane.showInputDialog("Enter a
number.");
        number = Integer.parseInt(input);

        // Determine whether it is prime or not.
        if (isPrime(number))
            message = "That is a prime number.";
        else
            message = "That is not a prime number.";

        // Display a message.
        JOptionPane.showMessageDialog(null, message);

        System.exit(0);
    }
}

```

Largest and Smallest Problem Code: P4

Write a program with a loop that lets the user enter a series of integers. The user should enter -99 to signal the end of the series. After all the numbers have been entered, the program should display the largest and smallest numbers entered.

Input

```
Enter an integer, or -99 to quit: 45
Enter an integer, or -99 to quit: 20
Enter an integer, or -99 to quit: 32
Enter an integer, or -99 to quit: 54
Enter an integer, or -99 to quit: 12
Enter an integer, or -99 to quit: -99
```

Expected Output:

```
Largest: 54
Smallest: 12
```

Solution: Largest and Smallest Problem Code: P4

```
package largestandsmallest;

import java.util.Scanner; // Needed for keyboard input

public class LargestAndSmallest {
    public static void main(String[] args)
    {
        String input;      // To hold keyboard input
        int num;            // A number
        int largest;        // The largest number
```

```

int smallest;           // The smallest number
// Create a Scanner object for keyboard input.
Scanner keyboard = new Scanner(System.in);
// Get the first number.
System.out.print("Enter an integer, or -99 to
quit: ");
num = keyboard.nextInt();
// Since this is the only number we have, it is
the both the
// largest and the smallest... so far.
largest = num;
smallest = num;
// Get the rest of the numbers.
while (num != -99)
{
    // Ask for another number.
    System.out.print("Enter an integer, or -99 to
quit: ");
    num = keyboard.nextInt();
    // Determine whether this is now the largest.
    if (num != -99 && num > largest)
        largest = num;
    // Determine whether this is now the smallest.
    if (num != -99 && num < smallest)
        smallest = num;
}

```

```
// Display the largest and smallest numbers.  
if (largest == -99)  
    System.out.println("You did not enter any  
numbers.");  
else  
{  
    System.out.println("Largest: " + largest);  
    System.out.println("Smallest: " + smallest);  
}  
}  
}
```

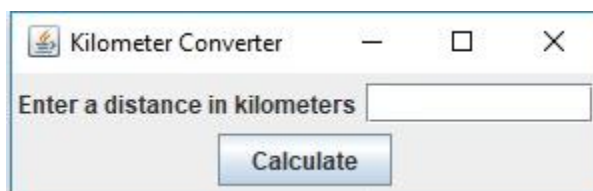
Kilometer Converter (GUI application) Problem Code: P5

Write a complete Java program that creates a window titled “Kilometer Converter” that is 300 by 100 pixels. The user can type a distance in kilometer into the text field. When the user presses a button labeled “Calculate”, the distance in kilometer converted to miles in Message dialog box.

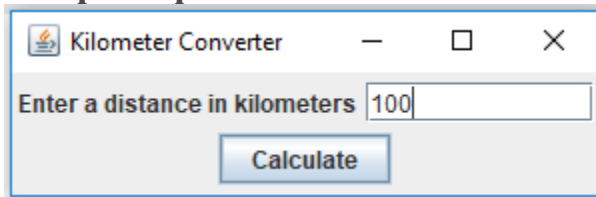
Note: The conversion formula is as follows:

Miles = Kilometers × 0.6214

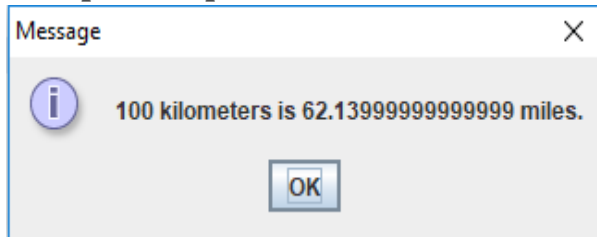
Your GUI might look like the following:



Sample Input:



Sample Output :



Solution: Kilometer Converter Problem Code: P5

```
package kiloconverter;

import javax.swing.*.*;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.BorderFactory;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import javax.swing.JButton;

import java.awt.event.*; // Needed for ActionListener
Interface

public class KiloConverter extends JFrame {
    private JPanel panel;           // To reference a
    panel
```

```

    private JLabel messageLabel;          // To reference a
label

    private JTextField kiloTextField; // To reference a
text field

    private JButton calcButton;          // To reference a
button

    private final int WINDOW_WIDTH = 310; // Window
width

    private final int WINDOW_HEIGHT = 100; // Window
height

    /**
    Constructor
    */

    public KiloConverter()
    {
        // Set the window title.
        setTitle("Kilometer Converter");

        // Set the size of the window.
        setSize(WINDOW_WIDTH, WINDOW_HEIGHT);
        setLocationRelativeTo(null);

        // Specify what happens when the close button is
clicked.
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // Build the panel and add it to the frame.
        buildPanel();
    }

```

```

        // Add the panel to the frame's content pane.
        add(panel);
        // Display the window.
        setVisible(true);
    }
    /**
        The buildPanel method adds a label, text field,
and
        and a button to a panel.
    */
    private void buildPanel()
    {
        // Create a label to display instructions.
        messageLabel = new JLabel("Enter a distance " +
                                   "in kilometers");
        // Create a text field 10 characters wide.
        kiloTextField = new JTextField(10);
        // Create a button with the caption "Calculate".
        calcButton = new JButton("Calculate");
        // Add an action listener to the button.
        calcButton.addActionListener(new
CalcButtonListener());
        // Create a JPanel object and let the panel
        // field reference it.
        panel = new JPanel();
        // Add the label, text field, and button

```

```

        // components to the panel.
        panel.add(messageLabel);
        panel.add(kiloTextField);
        panel.add(calcButton);
    }
    /**
     CalcButtonListener is an action listener class
for
     the Calculate button.
    */
    private class CalcButtonListener implements
ActionListener
    {
        /**
         The actionPerformed method executes when the user
         clicks on the Calculate button.
         @param e The event object.
        */
        public void actionPerformed(ActionEvent e)
        {
            final double CONVERSION = 0.6214;
            String input; // To hold the user's input
            double miles; // The number of miles

            // Get the text entered by the user into the
            // text field.

```

```

        input = kiloTextField.getText();

        // For debugging, display the text entered,
and
        // its value converted to a double.
        System.out.println("Reading " + input +
                           " from the text field.");
        System.out.println("Converted value: " +
                           Double.parseDouble(input));

        // Convert the input to miles.
        miles = Double.parseDouble(input) *
CONVERSION;

        // Display the result.
        JOptionPane.showMessageDialog(null, input +
        " kilometers is " + miles + "
miles.");

        // For debugging, display a message indicating
        // the application is ready for more input.
        System.out.println("Ready for the next
input.");
    }
} // End of CalcButtonListener class

/**

```

The main method creates an instance of the KiloConverter class, which displays its window on the screen.

```
*/
```

```
/**
```

```
    main method
```

```
*/
```

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

```
}
```