Activity 9 – Java Methods

Program 1 - This example demonstrates a program that calculates the total cost of items in a shopping cart. Type in the code and run the program.

```
public class ShoppingCart {
  // Method to calculate the total cost of items in the cart
  public static double calculateTotalCost(double[] prices) {
    double totalCost = 0.0;
    for (double price : prices) {
      totalCost += price;
    }
    return totalCost;
  }

  public static void main(String[] args) {
    double[] itemPrices = { 10.0, 15.0, 8.0, 20.0, 12.0 }; // Prices of individual items

    // Calculating the total cost of items in the cart
    double totalCost = calculateTotalCost(itemPrices);

    // Printing the total cost
    System.out.println("Total cost of items in the cart: £" + totalCost);
  }
}
```

In this example, we have a class named **ShoppingCart**. Inside this class, we have a method called **calculateTotalCost** which takes an array of prices as a parameter. It loops through the array and accumulates the prices to calculate the total cost.

In the main method, we define an array **itemPrices** representing the prices of individual items in the shopping cart. We then call the **calculateTotalCost** method with this array to find out the total cost.

Program 2 – This program calculates the area of different shapes using methods. Type in the code and run the program.

```
public class ShapeCalculator {

  // Method to calculate the area of a square
  public static double calculateSquareArea(double sideLength) {
    return sideLength * sideLength;
  }

  // Method to calculate the area of a rectangle
  public static double calculateRectangleArea(double length, double width) {
    return length * width;
  }

  // Method to calculate the area of a circle
  public static double calculateCircleArea(double radius) {
    return Math.PI * radius * radius;
  }

  public static void main(String[] args) {
    double sideLength = 5.0;
    double length = 6.0;
    double width = 4.0;
    double radius = 3.0;

    // Calculating areas of different shapes
    double squareArea = calculateSquareArea(sideLength);
    double rectangleArea = calculateRectangleArea(length, width);
    double circleArea = calculateCircleArea(radius);

    // Printing the results
```

```
        System.out.println("Area of the square: " + squareArea);

        System.out.println("Area of the rectangle: " + rectangleArea);

        System.out.println("Area of the circle: " + circleArea);

    }

}
```

In this example, we have a class named **ShapeCalculator**. Inside this class, we have three methods:

- **calculateSquareArea**: This method takes the length of a side and calculates the area of a square.
- **calculateRectangleArea**: This method takes the length and width of a rectangle and calculates its area.
- **calculateCircleArea**: This method takes the radius of a circle and calculates its area.

In the **main** method, we define different measurements for a square, a rectangle, and a circle. We then call the respective methods to calculate their areas.

Program 3 - This program contains a method that reverses a given string. Type in the code and run the program.

```
public class StringReverser {

  // Method to reverse a string
  public static String reverseString(String input) {
    StringBuilder reversed = new StringBuilder();
    for (int i = input.length() - 1; i >= 0; i--) {
      reversed.append(input.charAt(i));
    }
    return reversed.toString();
  }

  public static void main(String[] args) {
    String originalString = "Hello, World!";

    // Reversing the string
    String reversedString = reverseString(originalString);

    // Printing the result
    System.out.println("Original string: " + originalString);
    System.out.println("Reversed string: " + reversedString);
  }
}
```

In this example, we have a class named **StringReverser**. Inside this class, we have a method called **reverseString** which takes a String input and returns the reversed version of it.

The **reverseString** method uses a **StringBuilder** to build the reversed string by iterating through the characters of the input string in reverse order.

In the main method, we define a string **originalString** ("Hello, World!") and then call the **reverseString** method with this string.

Program 4 – Create a program to calculate the total cost of a list of products, each with its own price and quantity:

Hints:

Create a method called **calculateTotalCost** which takes three arrays as parameters: products (names of the products), prices (prices of the products), and quantities (quantities of each product). The method will then calculate the total cost by multiplying the price with the quantity for each product.

In the **main** method, define arrays for products, prices, and quantities. Call the **calculateTotalCost** method with these arrays.

Program 5 – Create a program to calculate the monthly salary of employees based on their hourly rate and hours worked

Hints:

Create a method called **calculateMonthlySalary** which takes two parameters: **hourlyRate** (the hourly rate of the employee) and **hoursWorked** (the number of hours worked in a week). The method will calculate the monthly salary (assuming 4 weeks in a month).

In the **main** method, define the hourly rates and hours worked for two employees. Call the **calculateMonthlySalary** method for each employee.

Program 6 – Change program 5 to enable as many employees to be entered as the user wishes. Also output the total hours worked and total salary (for all employees)