ASSIGNMENT (30-07-2024)

Write a program called SumProductMinMax3 that prompts user for three integers. The program shall read the inputs as int; compute the sum, product, minimum and maximum of the three integers; and print the results. For examples,

Enter 1st integer: 8

Enter 2nd integer: 2

Enter 3rd integer: 9

The sum is: 19

The product is: 144

The min is: 2

The max is: 9


```java
import java.util.Scanner;
public class SumProductMinMax3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter 1st integer: ");
        int num1 = scanner.nextInt();
        System.out.print("Enter 2nd integer: ");
        int num2 = scanner.nextInt();
        System.out.print("Enter 3rd integer: ");
        int num3 = scanner.nextInt();
        int sum = num1 + num2 + num3;
        int product = num1 * num2 * num3;
        int min = Math.min(num1, Math.min(num2, num3));
        int max = Math.max(num1, Math.max(num2, num3));
        System.out.println("The sum is: " + sum);
        System.out.println("The product is: " + product);
        System.out.println("The min is: " + min);
        System.out.println("The max is: " + max);
    }
}
```

Calculate BMI Using Java

The user enters his height (in inches) and weight (in pounds). The variables passed by the user are assigned to the float type. After calculating the BMI value, the value will be assigned to the appropriate range and the correct message will appear on the console. You can use the if-else-if ladder for printing the message on the console.

Intervals of BMI index:

16.00 or less = starvation

16.00-16.99 = emaciation

17.00-18.49 = underweight

18.50-22.99 = normal, low range

23.00-24.99 = normal high range

25.00-27.49 = overweight low range

27.50-29.99 = overweight high range

30.00-34.99 = 1st degree obesity

35.00-39.99 = 2nd degree obesity

40.00 or above = 3rd degree obesity

```java
import java.util.Scanner;
public class BMICalculator {
```

```java
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter your height in inches: ");
    float heightInInches = scanner.nextFloat();
    System.out.print("Enter your weight in pounds: ");
    float weightInPounds = scanner.nextFloat();
    float heightInMeters = heightInInches * 0.0254f;
    float weightInKilograms = weightInPounds * 0.453592f;
    float bmi = weightInKilograms / (heightInMeters * heightInMeters);
    System.out.println("Your BMI is: " + bmi);
    if (bmi <= 16.00) {
        System.out.println("Starvation");
    } else if (bmi <= 16.99) {
        System.out.println("Emaciation");
    } else if (bmi <= 18.49) {
        System.out.println("Underweight");
    } else if (bmi <= 22.99) {
        System.out.println("Normal, low range");
    } else if (bmi <= 24.99) {
        System.out.println("Normal, high range");
    } else if (bmi <= 27.49) {
        System.out.println("Overweight, low range");
    } else if (bmi <= 29.99) {
        System.out.println("Overweight, high range");
    } else if (bmi <= 34.99) {
        System.out.println("1st degree obesity");
    } else if (bmi <= 39.99) {
        System.out.println("2nd degree obesity");
    } else {
        System.out.println("3rd degree obesity");
    }
}
```

}


```
Output

java -cp /tmp/pTYtl8PDwC/BMICalculato
Enter your height in inches: 6
Enter your weight in pounds: 8
Your BMI is: 156.23756
3rd degree obesity


=== Code Execution Successful ===
```

Write a program that will use the while loop to find the largest and smallest number from the set of 10 randomly drawn integers from 1 to 100. In this task, do not use arrays or other collections.

import java.util.Random;

public class LargestSmallest {

   public static void main(String[] args) {

      Random random = new Random();

      int count = 1;

      int largest = Integer.MIN_VALUE;

      int smallest = Integer.MAX_VALUE;

      while (count <= 10) {

         int number = random.nextInt(100) + 1;

         System.out.println("Number " + count + ": " + number);

         if (number > largest) {

            largest = number;

         }

         if (number < smallest) {

            smallest = number;

         }

         count++;

      }

```java
        System.out.println("Largest number: " + largest);

        System.out.println("Smallest number: " + smallest);

    }

}
```

Output

```
java -cp /tmp/xKeqx4d4iU/LargestSmall
Number 1: 11
Number 2: 54
Number 3: 89
Number 4: 24
Number 5: 65
Number 6: 11
Number 7: 9
Number 8: 74
Number 9: 44
Number 10: 25
Largest number: 89
Smallest number: 9


=== Code Execution Successful ===
```