```
package com.itbulls.learnit.javacore.string;

import java.util.Formatter;

/**
 *
 * Specifiers, flags:
 * https://docs.oracle.com/javase/8/docs/api/java/util/Formatter.html
 *
 */

public class FormatterDemo {

	public static void main(String[] args) {
		// create Formatter class object
		Formatter formatter = new Formatter();

		// ================== Use Space format specifier
		System.out.println();
		System.out.println("===== Space format specifier =====");
		System.out.println();

		formatter.format("%d", -111);
		System.out.println(formatter);

		formatter = new Formatter();
		formatter.format("% d", 111);
		System.out.println(formatter);

		formatter = new Formatter();
		formatter.format("% d", -222);
```

```
System.out.println(formatter);

formatter = new Formatter();
formatter.format("% d", 222);
System.out.println(formatter);

// ================== + Sign Specifier
System.out.println();
System.out.println("===== + Sign Specifier =====");
System.out.println();

// + sign specifier
formatter = new Formatter();
formatter.format("%+d", 111);
System.out.println(formatter);

// + sign specifier
// on - sign, it will have no effect
formatter = new Formatter();
formatter.format("%+d", -111);
System.out.println(formatter);

// ================== ( specifier
System.out.println();
System.out.println("===== ( specifier =====");
System.out.println();

// ( Specifier
formatter = new Formatter();
formatter.format("%(d", -111);
System.out.println(formatter);
```

```
formatter = new Formatter();
formatter.format("%(d", 111);
System.out.println(formatter);

// ================== Comma, Specifier
System.out.println();
System.out.println("===== Comma, Specifier =====");
System.out.println();

// comma Specifier
formatter = new Formatter();
formatter.format("%, d", 1000000);
System.out.println(formatter);

// comma Specifier
formatter = new Formatter();
formatter.format("%, .3f", 32659526566.4521);
System.out.println(formatter);

// ================== Left Justification(-) Specifier
System.out.println();
System.out.println("===== Left Justification(-) Specifier =====");
System.out.println();

// right justify by default
formatter = new Formatter();
formatter.format("|%20.4f|", 1234.1234);
System.out.println(formatter);

// left justify
```

```
formatter = new Formatter();
formatter.format("|%-20.4f|", 1234.1234);
System.out.println(formatter);

// ================== The %n format specifiers
System.out.println();
System.out.println("===== The %n format specifiers =====");
System.out.println();

// newline format specifier
formatter = new Formatter();
formatter.format("Learn IT %nJava Courses %nby IT-Bulls");

// Print the output
System.out.println(formatter);

// ================== The %% format specifiers
System.out.println();
System.out.println("===== The %% format specifiers =====");
System.out.println();

formatter = new Formatter();

// %% format specifier
formatter.format("10 %% 4 = 2");

// Print the output
System.out.println(formatter);

// ================== The %x %X format specifiers
System.out.println();
```

```
System.out.println("===== The %x %X format specifiers =====");
System.out.println();

// %x format specifier
// It prints the number in Hexadecimal
// with lowercase alphabets
formatter = new Formatter();
formatter.format("%x", 250);

// Print the output
System.out.println("LowerCase Hexadecimal" + " using %x: " + formatter);

// %X format specifier
// It prints the number in Hexadecimal
// with uppercase alphabets
formatter = new Formatter();
formatter.format("%X", 250);

// Print the output
System.out.println("UpperCase Hexadecimal" + " using %X: " + formatter);

// ================== The %e %E format specifiers
System.out.println();
System.out.println("===== The %e %E format specifiers =====");
System.out.println();

// %e format specifier
// It prints the number in Scientific Notation
// with lowercase alphabets
formatter = new Formatter();
formatter.format("%e", 123.1234);
```

```
// Print the output
System.out.println("LowerCase Scientific Notation" + " using %e: " + formatter);

// %E format specifier
// It prints the number in Scientific Notation
// with uppercase alphabets
formatter = new Formatter();
formatter.format("%E", 123.1234);

// Print the output
System.out.println("UpperCase Scientific Notation" + " using %E: " + formatter);

// ================== Precision Formats
System.out.println();
System.out.println("===== Precision Formats =====");
System.out.println();

// added floating-point data
// using the %f or %e specifiers,
// Format to 2 decimal places
// in a 16 character field.
formatter = new Formatter();
formatter.format("%16.2e", 123.1234567);
System.out.println("Scientific notation to 2 places: " + formatter);

// Format 4 decimal places.
formatter = new Formatter();
formatter.format("%.4f", 123.1234567);
System.out.println("Decimal floating-point" + " notation to 4 places: " + formatter);
```

```
                // Format 4 places.
                // The %g format specifier causes Formatter
                // to use either %f or %e, whichever is shorter
                formatter = new Formatter();
                formatter.format("%.4g", 123.1234567);
                System.out.println("Scientific or Decimal floating-point " + "notation to 4 places: " +
formatter);

                // Display at most 15 characters in a string.
                formatter = new Formatter();
                formatter.format("%.15s", "12345678901234567890");
                System.out.println("String notation to 15 places: " + formatter);

                // Format into 10 digit
                formatter = new Formatter();
                formatter.format("%010d", 88);
                System.out.println("value in 10 digits: " + formatter);

        }

}
```