CS-207: Object-Oriented Programming & Data Structures

Northeastern Illinois University

Strings

Chapter 10

Practice Problem #1

- Create a class named `Decimals`.
- Write a `static` method named `printDecimalDigits` that takes a double as a parameter and does not return anything.
- The method should print out "`Digits after decimal point: `", followed by all the digits after the decimal point on one line (do not include the decimal point).
- Additional rule: Your code must not contain any loops.
- Create a test class named `DecimalsTest` that contains the `main` method and use the sample usage below to check your code.

Method Call	Method Output
`Decimals.printDecimalDigits(37.2)`	`Digits after decimal point: 2`
`Decimals.printDecimalDigits(1000)`	`Digits after decimal point: 0`
`Decimals.printDecimalDigits(18.3927493038)`	`Digits after decimal point: 3927493038`

CS-207: Programming II

Northeastern Illinois University

Strings

Chapter 10

Practice Problem #2

Create a class named `CaesarCipher` that has the following.

- Write a public static method named `encrypt` that takes a `String s` and an `int d` as parameters and returns a `String.`
- The Caesar cipher is one of the earliest known and simplest ciphers. It is a type of substitution cipher in which each letter in the text is "shifted" a certain number of places down the alphabet. For example, with a shift of 1, a would become b, b would become c, and so on. The method is named after Julius Caesar, who used it to communicate with his generals.
- The `encrypt` method should take the `String` parameter `s` and encrypt it using the Caesar cipher with a shift of d. Note that there is no upper limit (other than the max value of the `int` primitive type) on the value for d. You can assume that there is no punctuation and that the `String` parameter will always contain lower case letters.
- Additional rules: You may use one conditional (if) statements, and it can have only one (else) statement.
- Create a test class named `CaesarCipherTest` that contains the `main` method and prints the following method outputs in the following order, with a blank line between each encryption.

Method Call	Method Output
`CaesarCipher.encrypt("abc", 1)`	`Shift: 1` `abc encrypted is bcd`
`CaesarCipher.encrypt("abc", 25)`	`Shift: 25` `abc encrypted is zab`
`CaesarCipher.encrypt("abc", 50)`	`Shift: 50` `abc encrypted is yza`
`CaesarCipher.encrypt("attack at dawn", 17)`	`Shift: 17` `attack at dawn encrypted is rkkrtb rk urne`

- Write a public static method named `decrypt` that takes a `String s` and an `int d` as parameters and returns a `String`.
- The `decrypt` method should take the `String` parameter `s` and decrypt it using the Caesar cipher with a shift back of d. Note that there is no upper limit (other than the max value of the `int` primitive type) on the value for d. You can assume that there is no punctuation and that the `String` parameter will always contain lower case letters.

- Additional rules: You may use one conditional (if) statements, and it can have only one (else) statement.
- In your **CaesarCipherTest** class, print out the following method outputs in the following order with a blank space in between each decryption as well. Check the following usage examples to check your code.

Method Call	Method Output
`CaesarCipher.decrypt("bcd", 1)`	`Shift: 1` `bcd decrypted is abc`
`CaesarCipher.decrypt("zab", 25)`	`Shift: 25` `zab decrypted is abc`
`CaesarCipher.decrypt("yza", 50)`	`Shift: 50` `yza decrypted is abc`
`CaesarCipher.decrypt("rkkrtb rk urne", 17)`	`Shift: 17` `rkkrtb rk urne decrypted is attack at dawn`