Assignment #2 (CS210 – S24 – 10146)

# Submit the first 3 problems with a .docx file or an image file. See submission instructions for coding problems.

# Problem #1: (5 Points)

Convert the decimal number 157 to its binary, octal, and hexadecimal representations. Show your manual steps for each conversion.

# Problem #2: (5 Points)

Calculate the results of the following expressions with integer math of Java. *Do not use a calculator or computer program*. **Show your steps**.

1. 15 % (4 + 3 ^ 2) – 1
2. 7 – (8 / 3 – 1) \* 2
3. 3 \* (2 & 4) / (7 – 5)
4. ((2 + 3) \* 4) / (8 % 5)
5. 10 / (2 | 3) + 6 \* 4 – 3

# Problem 3: (10 points)

Assume the initial values of int x = 27 and int y = 4. Determine the values of x and y after executing each of the following statements. *Do not use a calculator or computer program*. **Show your steps**.

Each statement should be considered independently (they do not affect each other):

1. x += y;
2. x /= y;
3. x \*= y++;
4. x %= --y;
5. x = (x-- > ++y) ? y : x;
6. x = x << y;
7. x = x >> y;
8. x = x | y;
9. x = x & y;
10. x = x ^ y;

# Coding #1: **Caesar Cipher** Implementation (10 Points)

## **Objective**: Develop a Java program to implement a simple Caesar cipher for encryption of plain text using character shifting.

### **Part A: A Class with Encryption Method**

1. Method Specification:

* Method Name: encrypt
* Parameters:
  + String plainText - the text to be encrypted.
  + int shift - the number of positions each character in the text should be shifted.
* Return Type: String - the encrypted text.

1. Implementation Details:

* The method should shift each character in plainText by the number shift. Only encode characters in the English alphabet ('a' - 'z' and 'A' - 'Z'). Leave all other characters unchanged.
* A positive shift value means shifting to the right in the alphabet, while a negative shift means shifting to the left. For example, shifting “abc:xyz” by 3 will produce “def:abc” and shifting “Def-Abc” by -3 will produce “Abc-Xyz”.
* If shift exceeds 26, reduce it using modulo operation (shift % 26).
* Implement this method without using external libraries.

1. Unit Tests:

* Write a comprehensive set of unit tests to verify the functionality of the encrypt method under various scenarios including boundary conditions.

### **Part B: Application Entry Point**

1. A separate class with main method:

* Create a separate class that includes the main method.

1. Program Behavior:

* Read a line of text from standard input.
* Read an integer (the shift value) from standard input.
* Output the encrypted version of the input text on a single line using the encrypt method.
* The program should continue reading and processing input until all input lines are exhausted.
* *Do NOT prompt user for inputs*.

### **Submission Details**

* Ensure that your code is well-commented to describe the functionality.
* Verify that the program compiles and runs as expected before submission.
* Package all project files in a zip file with your name as part of the file name.

# Coding #2: **Prime Number Checker** (10 points)

## **Objective**: Develop a Java program to check if a given integer is a prime number.

### **Part A: A Class with Prime Checking Method**

1. Method Specification:

* Method Name: isPrime
* Parameters:
  + int number - the number to be checked for primality
* Return Type: Boolean - returns true if the number is prime, otherwise false.

1. Implementation Details:

* The method should determine whether number is prime.
* A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself.
* Consider edge cases such as numbers less than 2, which are not prime.
* Optimize the method to handle large numbers efficiently, ensuring that the solution is more efficient than checking all numbers up to number - 1.
* Implement this method without using external libraries.

1. Unit Tests:

* Write a comprehensive set of unit tests to verify the functionality of the isPrime method under various scenarios, including boundary conditions.

### **Part B: Application Entry Point**

1. A separate class with main method:

* Create a separate class that includes the main method.

1. Program Behavior:

* Read the integer from standard input.
* Use the isPrime method to determine if the entered integer is prime.
* Output a string on a line, either "Prime" or "Not prime" based on the result from the method.
* The program should continue to read and process additional integers until all inputs are exhausted.
* *Do NOT prompt user to enter an integer*.

### **Submission Details**

* Ensure that your code is well-commented to describe the functionality.
* Verify that the program compiles and runs as expected before submission.
* Package all project files in a zip file with your name as part of the file name.