DIRECTIONS:

- You are given the following problems or scenarios below.
- Your program must be unique, neat, efficient, and free from bugs or errors.
- Submit the whole **BlueJ Project** folder on or before **December 30** thru my email address: vgperdido@nvsu.edu.ph. Attach it as .zip file. Email **subject** should be: **Mid-Term-Project**.

BE UNIQUE! WORK INDEPENDENTLY! DO NOT CHEAT!

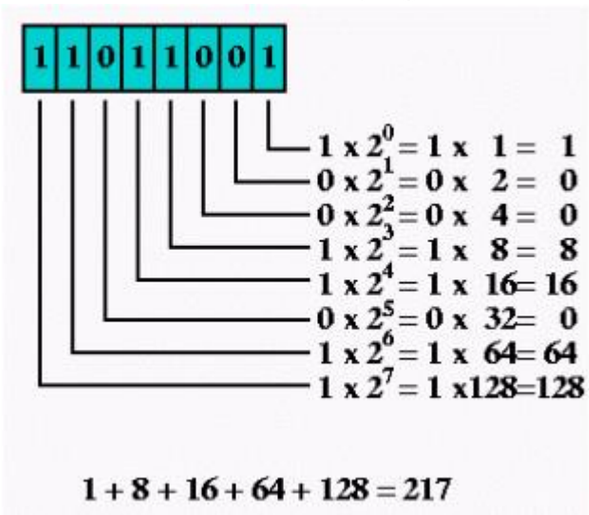
Subject:	**CSPC 3 (Object-Oriented Programming)**
Topics Covered:	Java Building Blocks, Built-in Methods, Control Structures, and Arrays
Intended Learning Outcome:	The students should be able to provide program solutions to machine problems involving the use of built-in methods available in the Java API. Also, students should be able to apply all the fundamental concepts of Java (variables, data types, operators, expressions) including the different types of control structures.

LAB ACTIVITY Description

1. Class name: BinaryToDecimalConverter

Binary number is passed as integer but we only consider its digits and not actual value. So input will always use 0 and 1. We are also not considering them as 2's complement number as Java do to represent negative integers in binary. Algorithm works by getting last digit in each iteration and then use the `Math.pow(2, position)` built-in method, where `2` is the binary base (constant) and `position` starts from zero. You can get the last digit of a number by using modulus operator e.g. `number % 10` will give you the last digit. The right most bit is known as first position and should be multiplied by 2 to the power zero i.e. 1. The loop continues till all digits are processed i.e. if input is 101 then it will run 3 times, if input is 1001 then it will four times. So it's complexity is `O(n)` because it will need n iteration to convert a n digit binary number into decimal. NOTE: Type cast the Math.pow() into `int` data type since it returns `double`.

Consider the following illustration on how to convert binary digit to decimal manually:



Note: Your program must check first if the binary digits is valid. All you need to do is to check every digit of number to see if they are greater than 1 or not. If any digit is greater than 1 then its not binary. For first timers challenge is how to write a loop to check every digit, well you need to remember one of the common tricks of programming. If you divide a number by 10 e.g. number/10, you reduce one digit from

it and if you use remainder operator e.g. `number % 10` then you will get last digit of number. For example `1234 / 10` will return 123 which means last digit 4 is removed and `1234 % 10` will return 4, which is the last digit. By using this two operators you can easily write a loop which can go through each digit and can check if its greater than 1 or not.

NOTE: Make a method to call if the user wants to try converting or not.

Sample Output Dialogue:

```
Enter binary digits (bits): 10111
The equivalent decimal value: 23

Try another conversion? Press Y/y to continue or any key to quit: Y

Enter binary digits (bits): 101201
Invalid binary digits!
```

2. Class Name: HexaToDecimal

Write a program called `Hex2Dec` to convert an input hexadecimal string into its equivalent decimal number. Your output shall look like:

```
Enter a Hexadecimal string: 1a
The equivalent decimal number for hexadecimal "1a" is 26

Enter a Hexadecimal string: 1y3
Error: Invalid Hexadecimal String "1y3"
```

Hints:

For a n-digit hexadecimal number $h_{n-1}h_{n-2}...h_1h_0$, $h_i \in \{0,\ldots,9,A,\ldots,F\}$, the equivalent decimal number is $h_{n-1}\times16^{n-1}+h_{n-2}\times16^{n-2}+ \ldots+h_1\times16^1+h_0\times16^0$.

You do not need a big nested-if statement of 16 cases (or 22 considering the upper and lower letters). Extract the individual character from the hexadecimal string, says `c`. If char `c` is between `'0'` to `'9'`, you can get the integer offset via `c-'0'`. If `c` is between `'a'` to `'f'` or `'A'` to `'F'`, the integer offset is `c-'a'+10` or `c-'A'+10`.

```
String hexStr;
char hexChar;
......
hexChar = hexStr.charAt(i);
......
if (hexChar >= '0' && hexChar <= '9') {
   ... (hexChar-'0') ...
   ...
} else if (hexChar >= 'a' && hexChar <= 'f') {   // lowercase
   ... (hexChar-'a'+10) ...
   ...
} else if (hexChar >= 'A' && hexChar <= 'F') {   // uppercase
   ... (hexChar-'A'+10) ...
   ...
} else {
   System.out.println("Error: Invalid hexadecimal string");
```

```
      System.exit(1);     // quit the program

      }
```

3. Class Name: GradesStatistics

Write a program called `GradesStatistics`, which reads in *n* grades (of `int` between `0` and `100`, inclusive) and displays the *average*, *minimum*, *maximum*, and *standard deviation*. Your program shall check for valid input. You should keep the grades in an `int[]` and use a method for each of the computations. Your output shall look like:

```
Enter the number of students: 4
Enter the grade for student 1: 50
Enter the grade for student 2: 51
Enter the grade for student 3: 56
Enter the grade for student 4: 53
The average is 52.5
The minimum is 50
The maximum is 56
The standard deviation is 2.29128784747792
```

Hints: The formula for calculating standard deviation is:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1} x_i^2 - \mu^2}, \text{ where } \mu \text{ is the mean}$$

```
public class GradesStatistics {
   public static int[] grades;  // Declare an int[], to be
allocated later

   // main() method
   public static void main(String[] args) {
      readGrades();
      System.out.println("The average is " + average());
      System.out.println("The minimum is " + min());
      System.out.println("The maximum is " + max());
      System.out.println("The standard deviation is " +
stdDev());
   }

   // Prompt user for the number of students and allocate the
"grades" array.
   // Then, prompt user for grade, check for valid grade, and
store in "grades".
   public static void readGrades() { ..... }

   // Return the average value of int[] grades
   public static double average() { ..... }

   // Return the maximum value of int[] grades
   public static int max() { ..... }

   // Return the minimum value of int[] grades
```

```
      public static int min() { ..... }

      // Return the standard deviation of the int[] grades
      public static double stdDev() { ..... }
   }
```

4. Class Name: GradesHistogram

Write a program called `GradesHistogram`, which reads in *n* grades (of `int` between `0` and `100`, inclusive) from a text file called "`grades.in`" and displays the histogram horizontally and vertically. The file has the following format:

```
numStduents:int
grade1:int grade2:int .... gradeN:int
```

For example:

```
15
49 50 51 59 0 5 9 10 15 19 50 55 89 99 100
```

The output shall consist of a horizontal histogram and a vertical histogram as follows:

```
  0 -  9: ***
 10 - 19: ***
 20 - 29:
 30 - 39:
 40 - 49: *
 50 - 59: *****
 60 - 69:
 70 - 79:
 80 - 89: *
 90 -100: **
```

Hints:

```
public class GradesHistogram {
   public static int[] grades;
      // Declare an int array of grades, to be allocated later
   public static int[] bins = new int[10];
      // Declare and allocate an int array for histogram bins.
      // 10 bins for 0-9, 10-19,...., 90-100

   public static void main(String[] args) {
      readGrades("grades.in");
      computeHistogram();
      printHistogramHorizontal();
      printHistogramVertical();
   }

   // Read the grades from "filename", store in "grades" array.
   // Assume that the inputs are valid.
   public static void readGrades(String filename) { ..... }
```

```
    // Based on "grades" array, populate the "bins" array.
    public static void computeHistogram() { ..... }

    // Print histogram based on the "bins" array.
    public static void printHistogramHorizontal() { ..... }

    // Print histogram based on the "bins" array.
    public static void printHistogramVertical() { ..... }
}
```

5. Class name: NumberGuess:

Write a program called `NumberGuess` to play the number guessing game. The program shall generate a random number between `0` and `99`. The player inputs his/her guess, and the program shall response with "Try higher", "Try lower" or "You got it in n trials" accordingly. For example:

```
Key in your guess:
50
Try higher
70
Try lower
65
Try lower
"
You got it in 4 trials!
```

GROUP NO.	GROUP MEMBERS
GROUP 1	ABBAGU,CHINO JOSHUA G. – **Leader** ADVINCULA, MARIA CHRISTINA AGGASID CHRISTINE KAYTE P. BERNARDEZ,ELCID B. BRIMON DEBBIE JANE B.
GROUP 2	DELA CRUZ ELIAN REI A. – **Leader** CABUENA GRANEL JAYLORD A CALDERON JHIMAR PADERO CUARIO,REYMARK.L. VELASCO, HAZEL D
GROUP 3	GARCIA, IAN PAUL A. – **Leader** PANIT, BRYCE JAMES R. SALAZAR, JERICK D. SOBREPENA, ELSA JANE A TORRES, KIMBLER JOHN D. TUMACDER, ALLYSSA DIANE L.
SUBJECT TEACHER	**Vilchor G. Perdido** Computer Science Department College of Arts and Sciences Nueva Vizcaya State University Email: vgperdido@nvsu.edu.ph