CSC 2262 Programming Project

Aymond, CSC 2262 Sections 1, Louisiana State University
Due: 11PM, Wednesday December 2

This program can be implemented in the programming language of your choice. Object-oriented programming is not required. Regardless of the programming language adopted, class style requirements must be adhered to.

Style Requirements

It is important to be a disciplined developer. We will practice discipline by adopting best practices in the style of all programs.

- Documentation/Comments
  - The "main" program/method should include comments that contain a descriptive overview of the program, the name of the developer, and the date of the assignment. For example:

```
/**
 * This program computes the volume (in liters) of a six-pack of
 * soda cans and the total volume of a six-pack of a two-liter
 * bottle.
 *
 * CSC 2262 Programming project No 3
 *
 * @author Type your name here
 * @since Type the date here, for example 9/10/2020
 *
 */
```

  - Each method//procedure/function should include a descriptive overview of the routine, the name of the developer, the date that the routine was developed and last edited, and a list of the parameters, to include datatype and a brief description. For example:

```
/**
 * This method computes the volume (in liters) of a pack of
 * soda cans.
 *
 * method: SodaPackVolume
 *
 * return type: double
 *
 * parameters:
 *    CanSize       [double]     the volume of the soda can, in
 *                               ounces
 *     PackSize     [int]        number of cans in pack
 *
 *
 * @author Type your name here
 * @since Type the date here, for example 9/10/2020
 *
```

```
*/
```

  - Except for trivial cases (e.g., variable used as loop control), all variable declarations should include a short comment that provides a brief description of the variable's purpose and/or use
  - Comments should be used liberally throughout code to document the programmer's intent
- Naming conventions
  - Programs and methods will be named using upper camel case (e.g., SetValue).
  - Variables will be named using lower camel case (e.g., firstValue)
  - Constants will be named using all caps and underscore to separate words (e.g., OUNCES_PER_CAN)
  - Member variables will start with "m_" followed by an upper camel case specification (e.g., m_IsOpen)
- Style conventions
  - Except for trivial cases (e.g., variable used as loop control), all variable declarations will appear at the top of a method
  - Each method will have exactly one entry point and exit point, unless use case dictates otherwise (i.e., not returns from a method call from within a logic structure)
  - Break, continue, and goto should not be used except when multiple selection statements or use case demands (e.g., exception handling in some programming languages)
- OO conventions
  - All member variables will be private, unless use case dictates otherwise
  - Class structure will follow the order
    - Public member variables (if any)
    - Private member variables
    - Private methods (if any)
    - Public methods

Assignment

Write a simple computer program that uses the Euler method to solve any ordinary differential equation.

Your program should output a table showing the values at each step in the evaluation interval The table should include a row header with the following entries: "h", "x", and "yh(x)"

Run your program using the following ordinary differential equation:

$y' = xy + \frac{4x}{y}$, $0 \leq x \leq 1$, $y(0) = 1$ with $h = 0.25$.

Example output only:

```
    h      x      yh(x)
 0.20   0.00     2.0000
 0.20   0.20     2.0000
```

```
0.20  0.40    2.0067
0.20  0.60    2.0305
0.20  0.80    2.0793
0.20  1.00    2.1592
0.20  1.20    2.2751
0.20  1.40    2.4310
0.20  1.60    2.6303
0.20  1.80    2.8757
0.20  2.00    3.1697
0.20  2.20    3.5143
0.20  2.40    3.9115
0.20  2.60    4.3627
0.20  2.80    4.8696
0.20  3.00    5.4332
0.20  3.20    6.0549
0.20  3.40    6.7356
0.20  3.60    7.4763
0.20  3.80    8.2779
0.20  4.00    9.1411
0.20  4.20   10.0668
0.20  4.40   11.0555
0.20  4.60   12.1079
0.20  4.80   13.2246
0.20  5.00   14.4062
0.20  5.20   15.6530
0.20  5.40   16.9657
0.20  5.60   18.3446
0.20  5.80   19.7902
0.20  6.00   21.3029
```

Submission

Your submission should include three things:

1. A capture the program output in a separate file for submission. This file can be a text file or a jpg.
2. In a separate file, provide a brief set of instructions that the grader should follow to compile and run your program.
3. Your program source code and any support files needed to run your program.

Zip all of these files into a single *.zip file for Moodle upload.