Computer Science – Object Programming in C++

Laboratory #3

Your task for today is to build a *FloatNumber* class, that will represent floating point value. In our solution the decimal part is bounded to MAX_DECIMAL_VALUE = 1000 (constant already defined in the program), which means that it consists up to three decimal digits.

After implementation of each stage uncomment appropriate section in Main.cpp file to test your code.

STAGE_1 (2 Points) – In this stage you have to implement two constructors with default values equals to zero and constant function WriteToConsole that writes the floatNumber to the console. Both constructors initialize integerValue and decimalValue fields inside the FloatNumber class.

Function WriteToConsole should output FloatNumber in following format:

- _ _ 5 . 5 4 3 – If float equals to 5.543f.
- _ 1 3 . * * 5 – If float equals to 13.005f.

Where _ is replaced with space and * is replaced with zero.

STAGE_2 (1 Point) – This stage consists of implementation of function ReadFromConsole, which prompts user to provide integerValue and decimalValue of FloatNumber class. The output should be as follow:

- Provide Integer Value: '4'
- Provide Decimal Value: '15'

Where numbers given within '' are data provided by user. The decimalValue couldn't be less than zero and greater than MAX_DECIMAL_VALUE, so prompt user for valid value if he prodives something invalid. Moreover the integerValue should be greater than zero.

STAGE_3 (1 Point) – In this stage you have to implement Equals function, which returns true if two numbers are equal and false if not. Function should be a constant function which is a member of our class and takes constant reference to other FloatNumber as argument.

STAGE_4 (1 Point) – In this stage you have to overload the '+' operator for out FloatNumber class. It should add two FloatNumber numbers with modulo addition of decimalValue part. After this operation if decimalValue is greater than MAX_DECIMAL_VALUE, then you have to increase integerValue:

- 1 5 . 5 5 0 + 2 . 2 1 8 = 1 7 . 7 6 8
- 1 5 . 5 5 0 + 2 . 4 6 9 = 1 8 . 0 1 9

STAGE_5 (1 Point) – In this stage another FloatNumber constructor is required. This constructor takes string of characters and converts it to our representation.

The output from the application that you have to achieve in your solution:

```
---------------------------   STAGE_1 (2Pts)   ---------------------------

Float Number:   0.000
Float Number:   2.000
Float Number:   4.099
Float Number:  13.007
Float Number:   2.250
Float Number:   7.050
Float Number:   3.999

---------------------------   STAGE_2 (1Pts)   ---------------------------

Provide Integer Value: 3
Provide Decimal Value: 045
Your Float Number:   3.045

---------------------------   STAGE_3 (1Pts)   ---------------------------

Float Number : 14.760 Not Equals To Float Number:   2.180
Float Number : 14.760 Equals To Float Number:  14.760

---------------------------   STAGE_4 (1Pts)   ---------------------------

Adds Float Numbers:   0.000 To Float Number:   0.029 With Result:   0.029
Adds Float Numbers:   2.000 To Float Number:   7.000 With Result:   9.000
Adds Float Numbers:   4.099 To Float Number:   4.099 With Result:   8.198
Adds Float Numbers:  13.007 To Float Number:  13.995 With Result:  27.002
Adds Float Numbers:   2.250 To Float Number:  18.000 With Result:  20.250
Adds Float Numbers:   7.050 To Float Number:   2.950 With Result:  10.000
Adds Float Numbers:   3.999 To Float Number:   9.998 With Result:  13.997

---------------------------   STAGE_5 (1Pts)   ---------------------------

Float Number From String:   4.550
Float Number From String:  64.080
```