Object Oriented Programming
Instructor: Natalia Chaudhry

Assignment 1: Basic class design

Deadline to submit the assignment is Tuesday 13th October, 2020 till 8 am

Submission instructions

1. Submit your file in a .zip or .rar format. Name your file with your Roll number (e.g. BSEF19M009.zip or BSEF19M009.rar)
2. Make a separate folder for each task of assignment (e.g. task1, task2…). All the headers and .cpp files must be in respective folder with a screenshot of the output of a program of the respective task.
3. Every Cpp file must contain your name and roll number(In comments) at the top of each program
4. Don't enclose your executable code in comments otherwise it will not be evaluated.

Instructions: (MUST READ)

- **No compensation or makeup assignment.**
- **Don't discuss with peers. Changing variable names/ changing for to while loop will not help you in hiding cheating attempt!**
- **You are not allowed to ask TA to verify/ prove your cheating case! Any such complaint from TAs will result in serious consequences. Don't expect any positive response from TAs in such regard.**
- **Cheating cases will result in deduction in sessionals.**
- **You are not allowed to consult Internet. Plagiarism cases will be strictly dealt.**
- **Queries are not allowed. Do whatever you are able to understand**

General instructions for all tasks: (marks will be deducted if the instructions are violated)

Note: All the programs should be implemented using class. You can take input in main() function and then call appropriate methods/ member functions of a designed class to set and get values. YOU MUST CREATE A SEPARATE CPP AND HEADER FILE(S) FOR CLASS DECLARATION AND DEFINITION.

• The attributes of class should be declared as **private** and member functions as **public**.

• All the member functions (expect constructor) should be declared inside the class and defined outside the class.

• You should not initialize the attributes while declaring them in class. The values should be assigned using member functions only. E.g. you cannot declare like:

```
Class Person
{
        Private:
        int age=25;
}
```

• The values should be initialized using a constructor. There must be a constructor in your defined class.

• All inputs should be taken in *main()* and all the final results should also be reported/ displayed in the main function.
• All the logic should be implemented in class' member functions. Main() should only input and output relevant values by calling relevant functions of the class.

Question 1: Marks: 10 marks

Develop a C++ class called *PrimeNumberGenerator*. An object of this class will generate successive prime numbers on request. Think of the object as being similar to the digital tasbeeh counter. It has a reset button which makes the counter return to 0. There is also a button to get the next prime number. On the tasbeeh, pressing this button increments the current value of the counter. In your object, pressing this button would generate the next prime number for display. You'll implement the reset button as a member function *reset()* and the other button as a function *getNextPrime()*. When an object is first created and the *getNextPrime()* function is called, it will return the first prime number, i.e., 2. Upon subsequent calls to the *getNextPrime()* function, the subsequent prime numbers will be returned. Define any others functions such as constructor and attributes that you need.

> According to Wikipedia: "a prime number is a natural number greater than 1, that has no positive divisors other than 1 and itself."

Expected Input/Output:

Enter 1 if you want to reset

Enter 2 if you want to get next prime number

Enter -1 to quit: 1

Counter value: 0

Enter 1 if you want to reset

And 2 if you want to get next prime number: 2

Counter value: 2

Enter 1 if you want to reset

And 2 if you want to get next prime number: 2

Counter value: 3

Enter 1 if you want to reset

And 2 if you want to get next prime number: 2

Counter value: 5

Enter 1 if you want to reset

And 2 if you want to get next prime number: -1

Question 2: Marks: 15 marks

Define a class called *StringFormatter*. The purpose of an object of this class is to store a string variable (you may use the C++ string type or a char array). An object of this class can be created by calling a constructor that accepts one string argument. The string to be passed as argument will be a long line of text, such as

"The world is indeed full of peril and in it there are many dark places. But still there is much that is fair. And though in all lands, love is now mingled with grief, it still grows, perhaps, the greater."

The object will also have a function called *printRightAligned()* which accepts one integer argument *n*. The value of the argument represents the maximum number of characters that can be displayed on a line. This function displays the string string stored in the object's attribute on the screen, right aligned and with no more than *n* characters per line. Similarly, there should be a function called *printLeftAlgigned()* which displays the text left aligned, again, with no more than *n* characters per line.

For *printRightAligned()*, you can assume that the display is 20 characters wide, i.e, if a line to be displayed contains 15 characters, you would display 5 spaces followed by the text.

Note: You are not allowed to use iomanip or any other library except string.

Expected input/ output:

Input string:

"The world is indeed full of peril and in it there are many dark places. But still there is much that is fair. And though in all lands, love is now mingled with grief, it still grows, perhaps, the greater."

For instance, if the *printRightAligned()* function is called with an argument value of 20, the following would be displayed on the screen:

```
The world is indeed
full of peril and in
and in it there are
many dark places.
But still there is
much that is fair.
And though in all
lands, love is now
mingled with grief,
```

it still grows,
perhaps, the greater.

Similarly, if the *printLeftAligned()* function were called with an argument value of 20, it would display the following text:

```
The world is indeed
full of peril and in
and in it there are
many dark places.
But still there is
much that is fair.
And though in all
lands, love is now
mingled with grief,
it still grows,
perhaps, the greater.
```