Object-Oriented Programming Language

Lab 1

The following are exercises that you will work on. If you find the problems very difficult, you might need to rethink taking this course.

1. Please create a class called HiNTU, and declare it in the HiNTU.h file. The following is the main program that you cannot change. Please define the `welcomToOOP` function in the HiNTU.cpp file.

```
#include "HiNTU.h"

int main(){
    ntu::HiNTU n;
    n.welcomeToOOP();

    return 0;
}
```

   And the following is the expected output:

```
Hello students of OOP!! Let's learn C++!!!
```

2. Please take an input from the user, and output whether the value is divisible by 2. The following are sample input and outputs (I/Os).

   Sample run 1:

   Output to user:

```
Please give an integer:
```

   User input:

```
1
```

   Final output:

```
The value is not divisible by 2.
```

   Sample run 2:

   Output to user:

```
Please give an integer:
```

   User input:

```
14
```

   Final output:

```
The value is divisible by 2.
```

3. Please output a pyramid of numbers. You will ask the height of the pyramid from the user, and output the corresponding pyramid.

   a. Numbers for all levels

   Sample run 1:

   Output to user:

```
Please give an integer:
```

   User input:

```
3
```

   Final output:

```
1
22
333
```

   Sample run 2:

   Output to user:

```
Please give an integer:
```

   User input:

```
5
```

   Final output:

```
1
22
333
4444
55555
```

   b. Numbers for odd levels, and * for even levels

   Sample run 1:

   Output to user:

```
Please give an integer:
```

   User input:

```
3
```

   Final output:

```
1
**
333
```

   Sample run 2:

   Output to user:

```
Please give an integer:
```

   User input:

```
5
```

Final output:

```
1
**
333
****
55555
```

4. In mathematics, the factorial of a non-negative value *n* is denoted as *n*!. For example, we say that factorial 5 is:

$$5! = 1 * 2 * 3 * 4 * 5$$

Please ask the user for the *n* value, and then output the corresponding factorial. In this practice, we do not consider the negative value of *n*. If such input is given from the user, please just return -1. The following are some example runs of the program.

Sample run 1:

Output to user:

```
Please give an integer:
```

User input:

```
3
```

Final output:

```
6
```

Sample run 2:

Output to user:

```
Please give an integer:
```

User input:

```
5
```

Final output:

```
120
```

Sample run 3:

Output to user:

```
Please give an integer:
```

User input:

```
-45
```

Final output:

```
Inpur should be non negative integer
```