Experiment No. 5
C++ CONDITIONAL STATEMENTS

I. OBJECTIVES

a.To implement the different conditional statements in C++ programs.
b.To be able to identify the application and limitations of the different conditional statement.

II. LABORATORY EXERCISES

Encode the following program and compile them. Illustrate the output.
Note: if getch(); will not work on other C++ ide compiler replace it with _getch();
For www.onlinegdb.com editor and cxxdroid (mobile app) you may remove it.

Program no.1 (save as ifex1.cpp)

```
#include<iostream>
#include<conio.h>
// this program will display a message if the entered number is 10
using namespace std;
int main ()
{

int a;

cout<<"\n"<<"Enter a number : " ;
cin>>a;
if (a==10)
cout<<a<<" is equal to 10. " ;
_getch();
return 0;
}
```

Program No. 2 (save as ifex2.cpp)

```
#include<iostream>
#include<conio.h>
using namespace std;
// this program will test if the number is positive or negative

int main ()

{
int a;
cout<<"\n"<<"Enter a number: ";
cin>>a;
if(a<0)
cout<<"The number is negative";
else
cout<<"The number is positive or zero: ";
_getch();
return 0;
}
```

Program No 3. (save as ifex3.cpp)

```
#include<iostream>
#include<conio.h>
using namespace std;
// this program will test if the number is positive, negative or zero

int main ()

{
int a;
cout<<"\n"<< "Enter a number : ";
cin>>a;
if(a<0)
cout<<"The number is negative.";
else if(a>0)
 cout<<"The number is positive.";
else
 cout<<"The number is zero";

_getch();

return 0;
}
```

Program No. 4(save as swex1.cpp)

```
#include<iostream>
#include<conio.h>
using namespace std;
int main ()
{
int code;
cout<<"Enter a code: " ;
cin>>code;
switch (code) {
            case 1: cout<<"\nFirst Year" ;
break;
case 2: cout<<"\nSecond Year" ;
break;
           case 3: cout<<"\nThird Year" ;
break;
case 4: cout<<"\nFourth Year" ;
break;
            case 5: cout<<"\nFifth Year" ;
break;
default:
                      cout<<"\nNot in the choices" ;
}
_getch();
return 0;
}
```

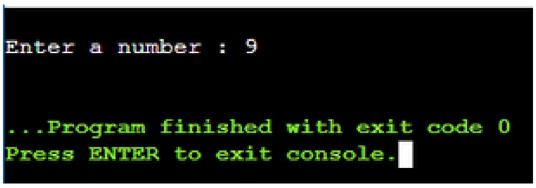
Program No. 5(save as prog5.cpp)

```
#include<iostream>
#include<conio.h>
using namespace std;
int main ()
{
char LtrGrade;
cout <<"Enter a letter grade: " ;
cin>> LtrGrade;
switch (LtrGrade) {
case 'a':
case 'A': cout<<"\nExcellent" ;
break;
case 'b':
case 'B': cout<<"\nSuperior" ;
break;
case 'c':
case 'C': cout<<"\nAverage" ;
break;
case 'd':
case 'D': cout<<"\nPoor" ;
break;
case 'f':
case 'F': cout<<"\nTry Again" ;
break;
default:
            cout<<"\nNo match was found for the ENTRY "<<LtrGrade<<endl;
}
_getch();
return 0;
}
```

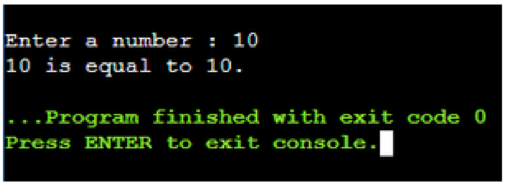
III. Summary of Program Outputs

Direction: Demonstrate the corresponding output for each of the given programs. Give your observation and analysis for each of the problems.

1. Because we only declared one condition which tests "if a is equals to 10", then we will display the value of a(if it is 10) and then the string "is equals to 10." .Otherwise, if we have input another value for a then it will not display anything because it doesn't have a default directive. Thus if we enter any number than 10 it will appear as blank.

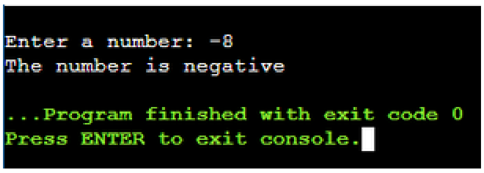


```
Enter a number : 9


...Program finished with exit code 0
Press ENTER to exit console.
```

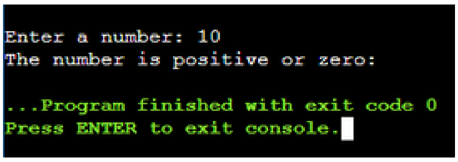


```
Enter a number : 10
10 is equal to 10.

...Program finished with exit code 0
Press ENTER to exit console.
```

2. In this code, we only tested it if it was a negative number by comparing a variable's value if it is less than 0 (a<0) and if the condition is met, it will say that the number is negative. Otherwise it will output that the number is either positive or zero, since we didn't type any condition to identify if the number input is Zero or Positive.

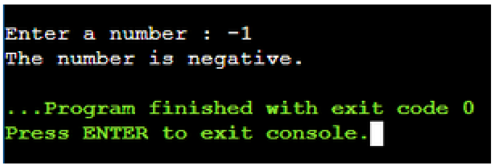


```
Enter a number: -8
The number is negative

...Program finished with exit code 0
Press ENTER to exit console.
```

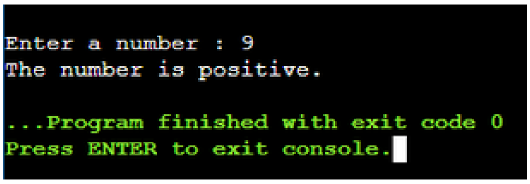


```
Enter a number: 10
The number is positive or zero:

...Program finished with exit code 0
Press ENTER to exit console.
```

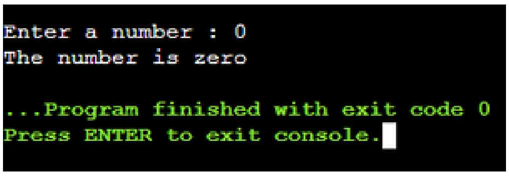
3. Now in this code, we wrote a code that tested if the value of "a" is greater than or less than 0, If it is less than zero(a<0) it will display that the number is negative. And if it is greater than zero(a>0) then it will display that the number is positive. And then we used a default directive that if none of the two conditions are met it will display that the number is zero, we accomplished this by using an if else statement by using conditions that will test the value of the variable.



```
Enter a number : -1
The number is negative.

...Program finished with exit code 0
Press ENTER to exit console.
```



```
Enter a number : 9
The number is positive.

...Program finished with exit code 0
Press ENTER to exit console.
```

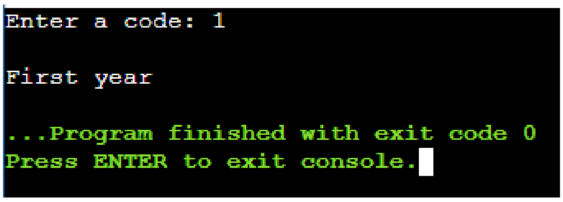


```
Enter a number : 0
The number is zero

...Program finished with exit code 0
Press ENTER to exit console.
```

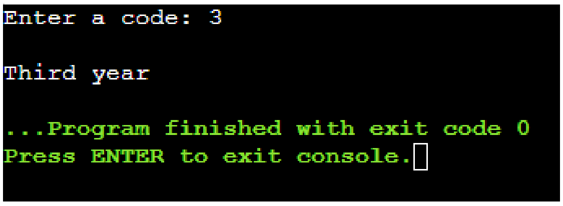
III. Summary of Program Outputs

Direction: Demonstrate the corresponding output for each of the given programs. Give your observation and analysis for each of the problems.

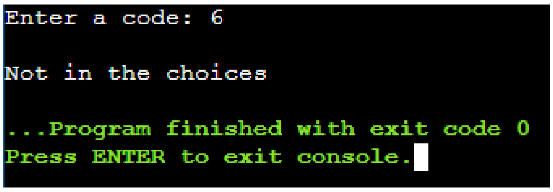
4. Unlike the if-else statement, switch statements test the given input and if it meets a specific case it will do the given objectives under that case. the cases in switch statements are exact value, range or decimal cannot be accepted, but multiple cases can be used under one directive output. and it also has a default directive like If-Else statements. In this code, we prompt the user to input a number that corresponds to the equivalent year level up to 5th year. And if the number input is not equal to the any of the cases, then it will go to the default directive "Not in the choices".



```
Enter a code: 1

First year

...Program finished with exit code 0
Press ENTER to exit console.
```

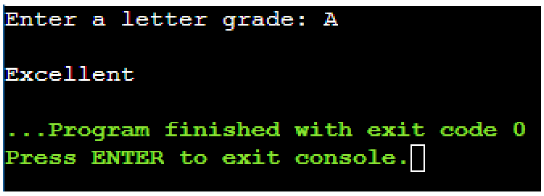


```
Enter a code: 3

Third year

...Program finished with exit code 0
Press ENTER to exit console.
```

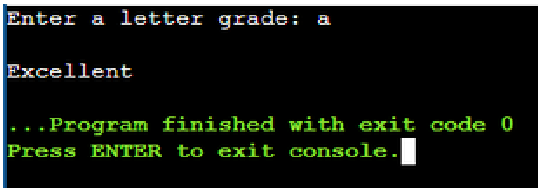


```
Enter a code: 6

Not in the choices

...Program finished with exit code 0
Press ENTER to exit console.
```

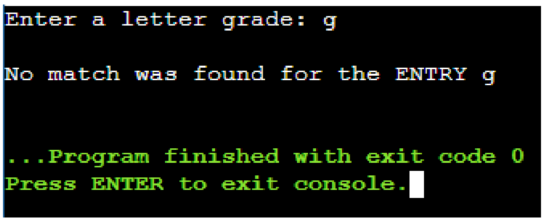
5. Just like what I've said earlier, multiple cases can be used under one directive output, just like here, we used two different cases for one directive output one Uppercase letter and one Lowercase letter. First we prompt the user to enter a letter grade, and then we used a switch statement to figure out if the given letter grade has met any of the cases. And if it doesn't meet any of the cases it will go directly to the default directive.



```
Enter a letter grade: A

Excellent

...Program finished with exit code 0
Press ENTER to exit console.
```



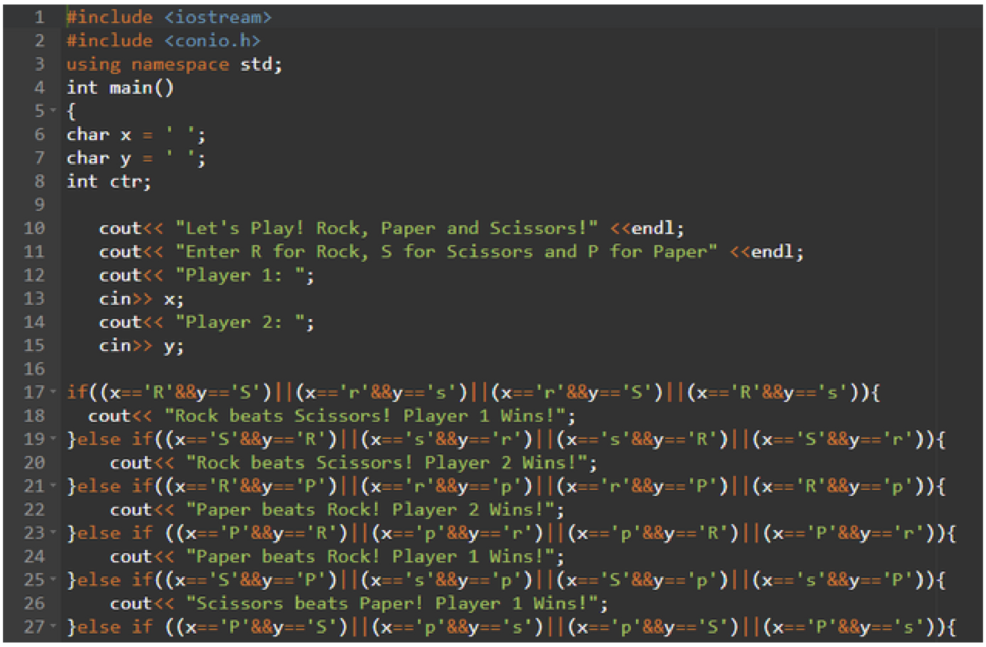
```
Enter a letter grade: a

Excellent

...Program finished with exit code 0
Press ENTER to exit console.
```



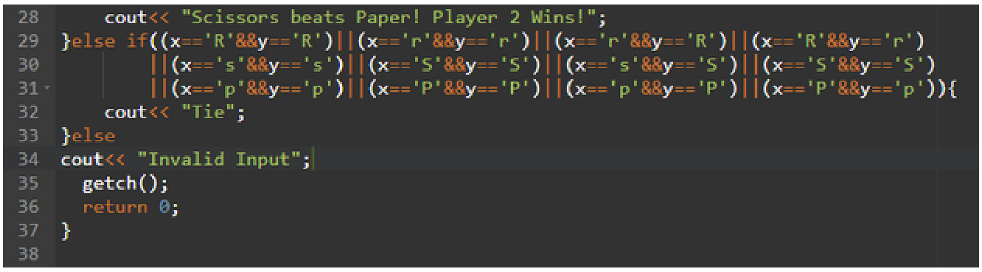
```
Enter a letter grade: g

No match was found for the ENTRY g


...Program finished with exit code 0
Press ENTER to exit console.
```

IV. Supplementary Problems

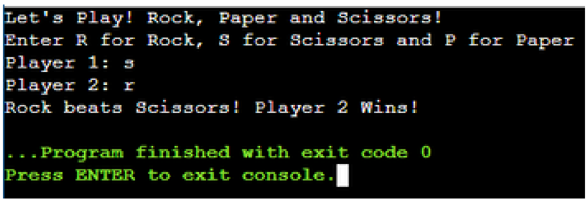
1.Write a “scissor-rock-paper” game, commonly known as “jack n poy” in the Philippines. Two players will input the first equivalent letter. For example, s for scissor, r for rock, and p for paper. Then display the winner and the basis of winning such as: “paper covers rock”, “rock breaks scissors”, and “scissors cut paper”, or “nobody wins”. Capital or small letters should be accepted by your program.



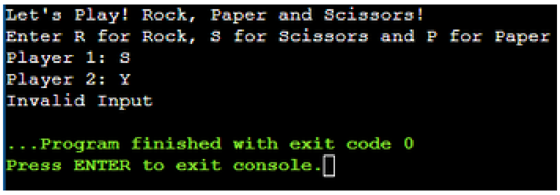
```cpp
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
char x = ' ';
char y = ' ';
int ctr;

    cout<< "Let's Play! Rock, Paper and Scissors!" <<endl;
    cout<< "Enter R for Rock, S for Scissors and P for Paper" <<endl;
    cout<< "Player 1: ";
    cin>> x;
    cout<< "Player 2: ";
    cin>> y;

if((x=='R'&&y=='S')||(x=='r'&&y=='s')||(x=='r'&&y=='S')||(x=='R'&&y=='s')){
  cout<< "Rock beats Scissors! Player 1 Wins!";
}else if((x=='S'&&y=='R')||(x=='s'&&y=='r')||(x=='s'&&y=='R')||(x=='S'&&y=='r')){
    cout<< "Rock beats Scissors! Player 2 Wins!";
}else if((x=='R'&&y=='P')||(x=='r'&&y=='p')||(x=='r'&&y=='P')||(x=='R'&&y=='p')){
    cout<< "Paper beats Rock! Player 2 Wins!";
}else if ((x=='P'&&y=='R')||(x=='p'&&y=='r')||(x=='p'&&y=='R')||(x=='P'&&y=='r')){
    cout<< "Paper beats Rock! Player 1 Wins!";
}else if((x=='S'&&y=='P')||(x=='s'&&y=='p')||(x=='S'&&y=='p')||(x=='s'&&y=='P')){
    cout<< "Scissors beats Paper! Player 1 Wins!";
}else if ((x=='P'&&y=='S')||(x=='p'&&y=='s')||(x=='p'&&y=='S')||(x=='P'&&y=='s')){
```



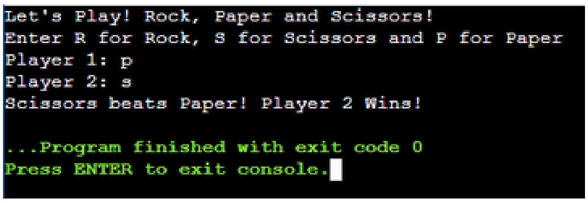
```cpp
    cout<< "Scissors beats Paper! Player 2 Wins!";
}else if((x=='R'&&y=='R')||(x=='r'&&y=='r')||(x=='r'&&y=='R')||(x=='R'&&y=='r')
        ||(x=='s'&&y=='s')||(x=='S'&&y=='S')||(x=='s'&&y=='S')||(x=='S'&&y=='S')
        ||(x=='p'&&y=='p')||(x=='P'&&y=='P')||(x=='p'&&y=='P')||(x=='P'&&y=='p')){
    cout<< "Tie";
}else
cout<< "Invalid Input";
  getch();
  return 0;
}
```



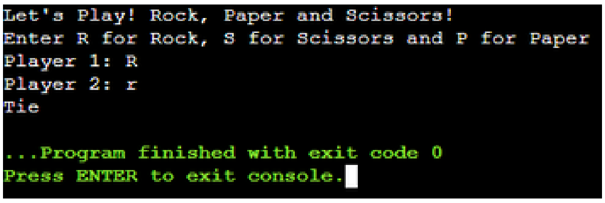
```
Let's Play! Rock, Paper and Scissors!
Enter R for Rock, S for Scissors and P for Paper
Player 1: s
Player 2: r
Rock beats Scissors! Player 2 Wins!

...Program finished with exit code 0
Press ENTER to exit console.
```



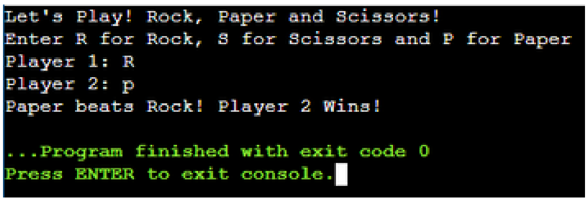
```
Let's Play! Rock, Paper and Scissors!
Enter R for Rock, S for Scissors and P for Paper
Player 1: S
Player 2: Y
Invalid Input

...Program finished with exit code 0
Press ENTER to exit console.
```



```
Let's Play! Rock, Paper and Scissors!
Enter R for Rock, S for Scissors and P for Paper
Player 1: p
Player 2: s
Scissors beats Paper! Player 2 Wins!

...Program finished with exit code 0
Press ENTER to exit console.
```



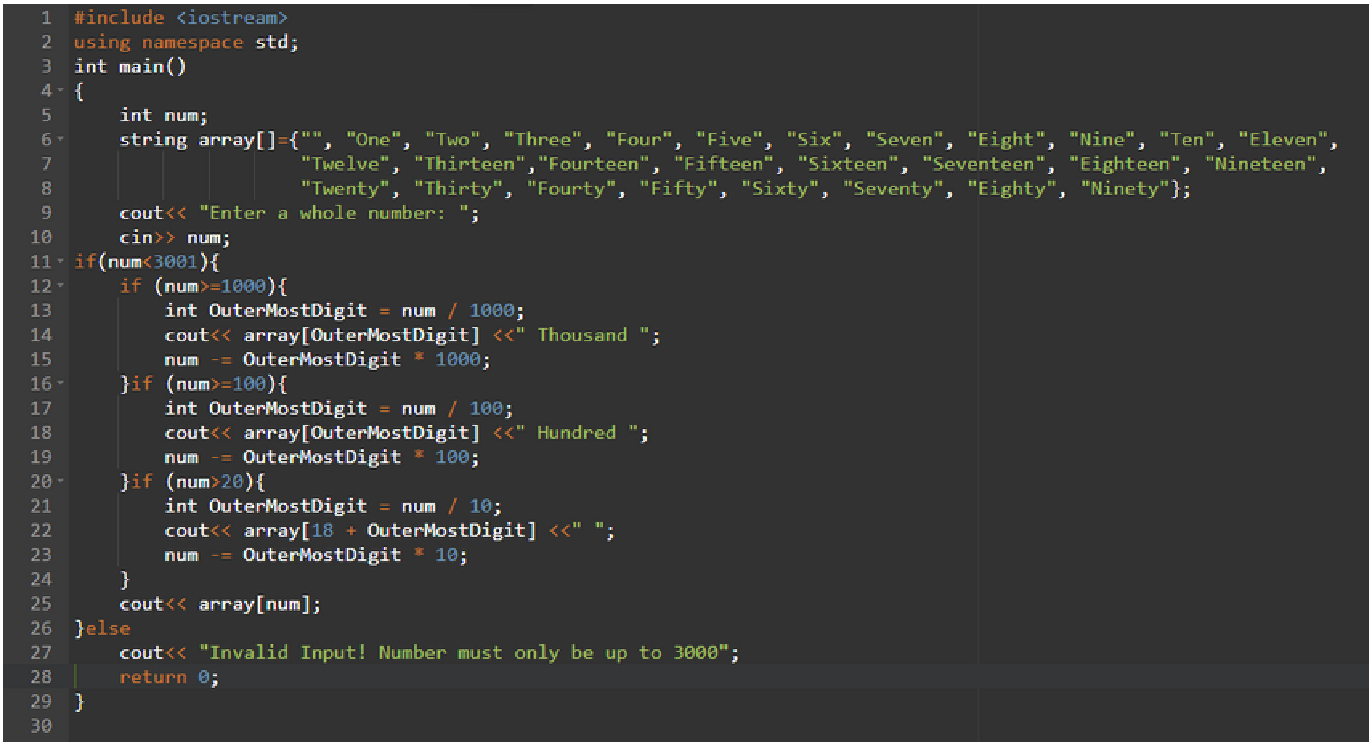
```
Let's Play! Rock, Paper and Scissors!
Enter R for Rock, S for Scissors and P for Paper
Player 1: R
Player 2: r
Tie

...Program finished with exit code 0
Press ENTER to exit console.
```



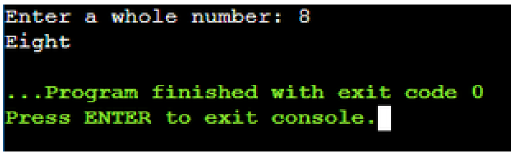
```
Let's Play! Rock, Paper and Scissors!
Enter R for Rock, S for Scissors and P for Paper
Player 1: R
Player 2: p
Paper beats Rock! Player 2 Wins!

...Program finished with exit code 0
Press ENTER to exit console.
```

IV. Supplementary Problems

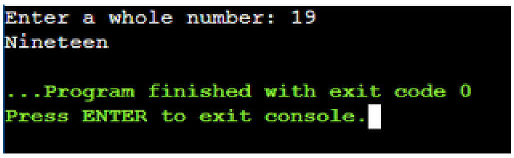
2.Write a program that accepts an ordinary number and outputs its equivalent in words. For example the inputted number is 1380 the output will be “one thousand three hundred eighty”. Take note that the maximum input number is 3000.



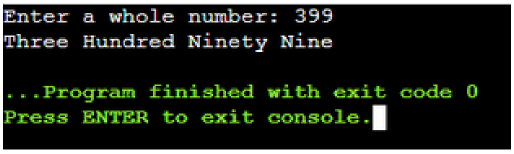
```
#include <iostream>
using namespace std;
int main()
{
    int num;
    string array[]={"", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten", "Eleven",
                    "Twelve", "Thirteen","Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen",
                    "Twenty", "Thirty", "Fourty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"};
    cout<< "Enter a whole number: ";
    cin>> num;
if(num<3001){
    if (num>=1000){
        int OuterMostDigit = num / 1000;
        cout<< array[OuterMostDigit] <<" Thousand ";
        num -= OuterMostDigit * 1000;
    }if (num>=100){
        int OuterMostDigit = num / 100;
        cout<< array[OuterMostDigit] <<" Hundred ";
        num -= OuterMostDigit * 100;
    }if (num>20){
        int OuterMostDigit = num / 10;
        cout<< array[18 + OuterMostDigit] <<" ";
        num -= OuterMostDigit * 10;
    }
    cout<< array[num];
}else
    cout<< "Invalid Input! Number must only be up to 3000";
    return 0;
}
```



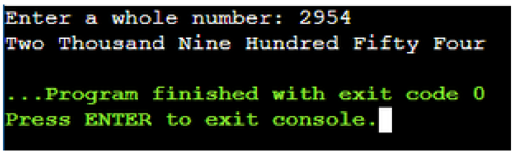
```
Enter a whole number: 8
Eight

...Program finished with exit code 0
Press ENTER to exit console.
```



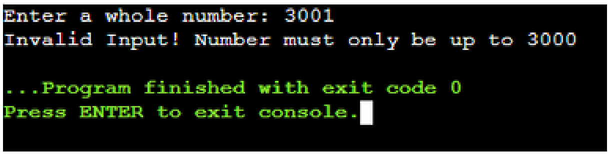
```
Enter a whole number: 19
Nineteen

...Program finished with exit code 0
Press ENTER to exit console.
```



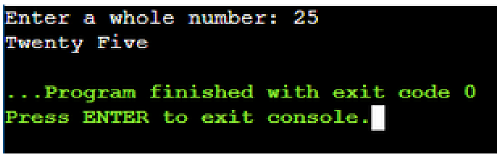
```
Enter a whole number: 399
Three Hundred Ninety Nine

...Program finished with exit code 0
Press ENTER to exit console.
```



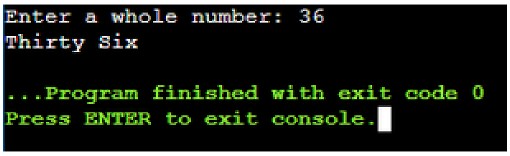
```
Enter a whole number: 2954
Two Thousand Nine Hundred Fifty Four

...Program finished with exit code 0
Press ENTER to exit console.
```



```
Enter a whole number: 3001
Invalid Input! Number must only be up to 3000

...Program finished with exit code 0
Press ENTER to exit console.
```



```
Enter a whole number: 25
Twenty Five

...Program finished with exit code 0
Press ENTER to exit console.
```



```
Enter a whole number: 36
Thirty Six

...Program finished with exit code 0
Press ENTER to exit console.
```

V. Conclusion

In conclusion, This activity taught me how to write conditional statements in C++, by the use of If, If-Else and Switch statements. As a future Computer Engineer, we should be well versed in using this concepts to solve problems or create codes that will solve specific problems. Some of the problems that I have encountered during this Experiment was, setting the condition for the Rock-Paper-Scissors game, I tried multiple times on setting the condition, but what I wrote seems to be an initial Logical Error, it would sometimes output what I didn't have in mind, but by reviewing my lessons on the conditional statement, I seem to notice that the conditions that I have set was to complex, making the compiler confused on what I was asking it. Nevertheless, it was still a fun experience as I learned new things that I can proudly say made programming fun.