OBJECT: To understand Relational Operators and Loops

RELATIONAL OPERATORS SUMMARY

Two expressions can be compared using relational and equality operators. For example, to know if two values are equal or if one is greater than the other. The result of such an operation is either true or false (i.e., a Boolean value).

>	Greater Than	<	Less Than
==	Equal To	!=	Not Equal To
>=	Greater Than or Equal To	<	Less Than or Equal To

THE FOR LOOP

```
for ( initialization ; Test Condition ; increment or decrement )
{
Body of the Loop;
}
```

This loop runs as long as the condition in the center is true. Note that there is no semi colon after the "for" statement. If there is only one statement in the "for" loop then the braces may be removed. If we put a semicolon after the for loop instruction then that loop will not work for any statement.

THE WHILE LOOP

```
while ( condition is true )
{
Body of the Loop;
}
```

This loop runs as long as the condition in the parenthesis is true. Note that there is no semicolon after the "while" statement. If there is only one statement in the "while" loop then the braces may be removed.

THE DO-WHILE LOOP

```
do
{
Body of the Loop;
}
while(condition is true);
```

This loop runs as long as the condition in the parenthesis is true. Note that there is a semicolon after the "while" statement. The difference between the "while" and the "do-while" statements is that in the "while" loop the test condition is evaluated before the loop is executed, while in the "do" loop the test condition is evaluated after the loop is executed. This implies that statements in a "do" loop are executed at least once. However, the statements in the "while" loop are not executed if the condition is not satisfied.

PROGRAM 1: Demonstrating the simple for loop

```cpp
#include <iostream>
using namespace std;
int main()
{

for (int k = 1 ; k <= 10 ; k++ )
cout << "Name" <<endl;

return 0;
}
```

PROGRAM 2: Demonstrating the simple for loop

```cpp
#include <iostream>
using namespace std;
int main()
{

for (int k = 1 ; k <= 1000 ; k+=10 )
cout << k << " ";

return 0;
}
```

PROGRAM 3: Demonstrating the simple for loop

```cpp
#include <iostream>
using namespace std;
int main()
{

for (int k = 10 ; k >= 0 ; k-- )
cout << k << " ";

return 0;
}
```

PROGRAM 4: Demonstrating the simple for loop

```cpp
#include <iostream>
using namespace std;
int main()
{

for (int k = 1 ; k <= 10000 ; k = (k+k)*10 )
cout << k << " ";

return 0;
}
```

PROGRAM 5: Demonstrating the simple for loop

```cpp
#include <iostream>
using namespace std;
int main()  {
int k, num;
cout << "Enter Table Number" ;    cin >> num;
for (k = 1 ; k <= 10 ; k++ )
cout << num << " * " << k << " = " << num*k << endl;
return 0;
}
```

PROGRAM 6: Demonstrating the simple for loop

```cpp
#include <iostream>
using namespace std;
int main()  {
int m, k = 0;
for ( ; k <= 10 ; )
{
cout << "Enter Marks" ;  cin >> m;
cout << "Marks = " << m;
k++;
}
return 0;
}
```

PROGRAM 7: Demonstrating the nested for loops

```cpp
#include <iostream>
using namespace std;
int main() {
const int N = 5;
int i,j;
for (i=1; i<=N; i++)
{
   for (j=1; j<=i; j++)
   cout << "* ";
   cout << endl;
   }

for (i=N; i>=1; i--)
{
   for (j=1; j<=i; j++)
   cout << "* ";
   cout << endl;
}

return 0;
}
```

PROGRAM 8: Demonstrating the simple while loop

```cpp
#include <iostream>
#include <cmath>
#include<conio.h>
using namespace std;

int main() {
char ch = 'y';
double number, answer;
while (ch == 'y')
{
cout << "Enter a number: ";
cin >> number;
answer = sqrt(number);
cout << "Square root is "<< answer << endl;
cout << "Do you want the square root of another number?" << endl;
cout << "Press y for yes... " << endl << endl;
ch = getche();
}
return 0;
}
```

PROGRAM 9: Demonstrating the simple do-while loop

```cpp
#include <iostream>
#include <cmath>
#include<conio.h>
using namespace std;

int main() {
char ch;
double number, answer;
do
{
cout << "Enter a number: ";
cin >> number;
answer = sqrt(number);
cout << "Square root is "<< answer << endl;
cout << "Do you want the square root of another number?" << endl;
cout << "Press y for yes... " << endl << endl;
ch = getche();
}
while (ch == 'y');
return 0;
}
```

Exercise 1:
Write a program that prints the values by using for loop

1	5	25	125	625	3125	15625	78125	390625	1953125

Exercise 2:
Write a program that computes the nth Fibonacci number where n is a value input by the user.

```
Enter N: 10
Fibonacci series till 10th place:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34
```

Exercise 3:
Write a program to find the factorial of a number.
The factorial function (symbol: !) says to multiply all whole numbers from our chosen number down to 1.
Examples:
4! = 4 × 3 × 2 × 1 = 24
7! = 7 × 6 × 5 × 4 × 3 × 2 × 1 = 5040
1! = 1

The program should have following output:

```
Enter a number to calculate Factorial:  9
9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1 = 362880
```

Exercise 4:
Write a program which takes a 3 numbers from users
1. Table number
2. Starting values of table
3. Ending value of table

Then it displays the table of the given number from starting value till ending value. For Example:

```
Enter Table Number: 7
Enter Starting Value: 6
Enter Ending Value: 12

7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
7 x 11 = 77
7 x 12 = 84
```