```
// Author - Krishna Kumari

// PRE/POST INCREMENT/DECREMENT OPERATORS


#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    int b = 20;
    int c = 30;
    int d = 40;


    // PRE INCREMENT -> First inc then use
    cout << ++a << endl;  // 11
    cout << (++a) * 5 << endl; //60

    // PRE DECREMENT -> First dec then use
    cout << (--b)  << endl;  // 19
    cout << (--b) * 5 << endl; // 90

    // POST INCREMENT -> First use then inc
    cout << (c++)  << endl;  // 30
    cout << (c++) * 5 << endl;  // 155

    // POST DECREMENT -> First use then dec
    cout << (d--)  << endl;  // 40
    cout << (d--) * 5 << endl;  // 195
    return 0;
}


//-----------------------------------------------

// BREAK & CONTINUE STATEMENTS

/*Break - The break statement is used to terminate the loop. If we want
to terminate the loop based on some conditions , then we use break statement. */


#include <iostream>
using namespace std;
int main()
{
    int i;
    for( i = 0; i < 10; i++)
    {
        if(i==5)
        {
            break;
        }
        cout << i <<" " ;
    }
    cout << endl;
    cout << "Loop terminate at i = " << i << endl;
    return 0;
}
```

```
//-----------------------------------------------

/* Continue -> is used to skip the current iteration of the loop.
If we want to skip any iteration based on some conditions, then we use
continue statement*/

#include <iostream>
using namespace std;
int main()
{
    int i;
    for( i = 0; i < 10; i++)
    {
        if(i==5)
        {
            continue;
        }
        cout << i <<" " ;
    }
    cout << endl;
    cout << "Iteration skips at i = 5 ";
    return 0;
}


//-----------------------------------------------\


// VARIABLE SCOPING  --> 1] local variable  2] Global variable

//Local variable are accessible and updated in its scope only//


/* for(int int i = 0; i < n; i++) // i is local variable
{
    cout << i << endl;
}
*/



/* Global variable are accessible anywhere in the program. These variable
   are intialized or declared inside or outside the main function generally.
   But Global variable are considered as bad practice.
 */


#include <iostream>
using namespace std;

int a = 12;
int main()
{
    int b=10;
    int c;
    c = a+b;
    cout << c ;
    return 0;
}
```

```
/* Note: 1] We can update a variable but can't redeclare it.
         2] We can declare the variable inside the nested STATEMENTS
         if-else or loops.
*/
```