LAB-05

SAJJAD ALI RAJPER

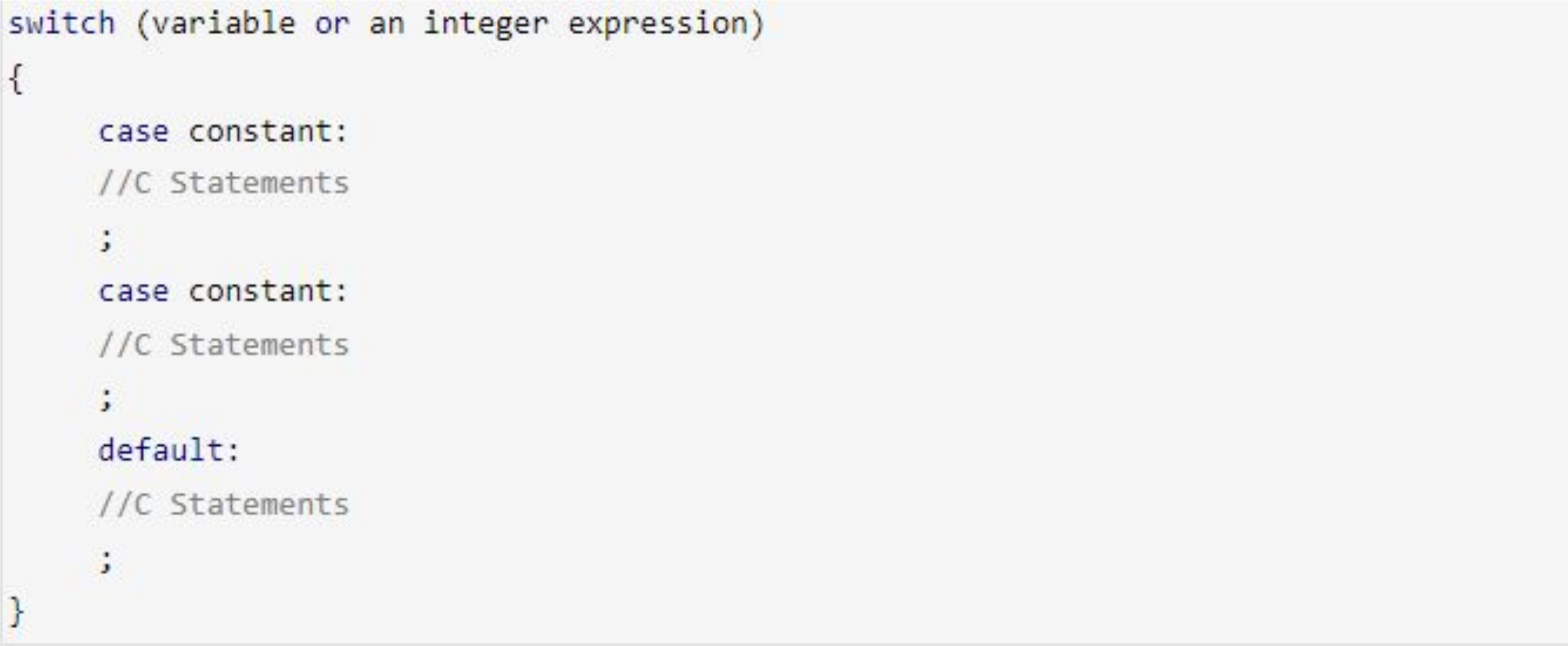
LECTURER

AROR UNIVERSITY

Switch case statement

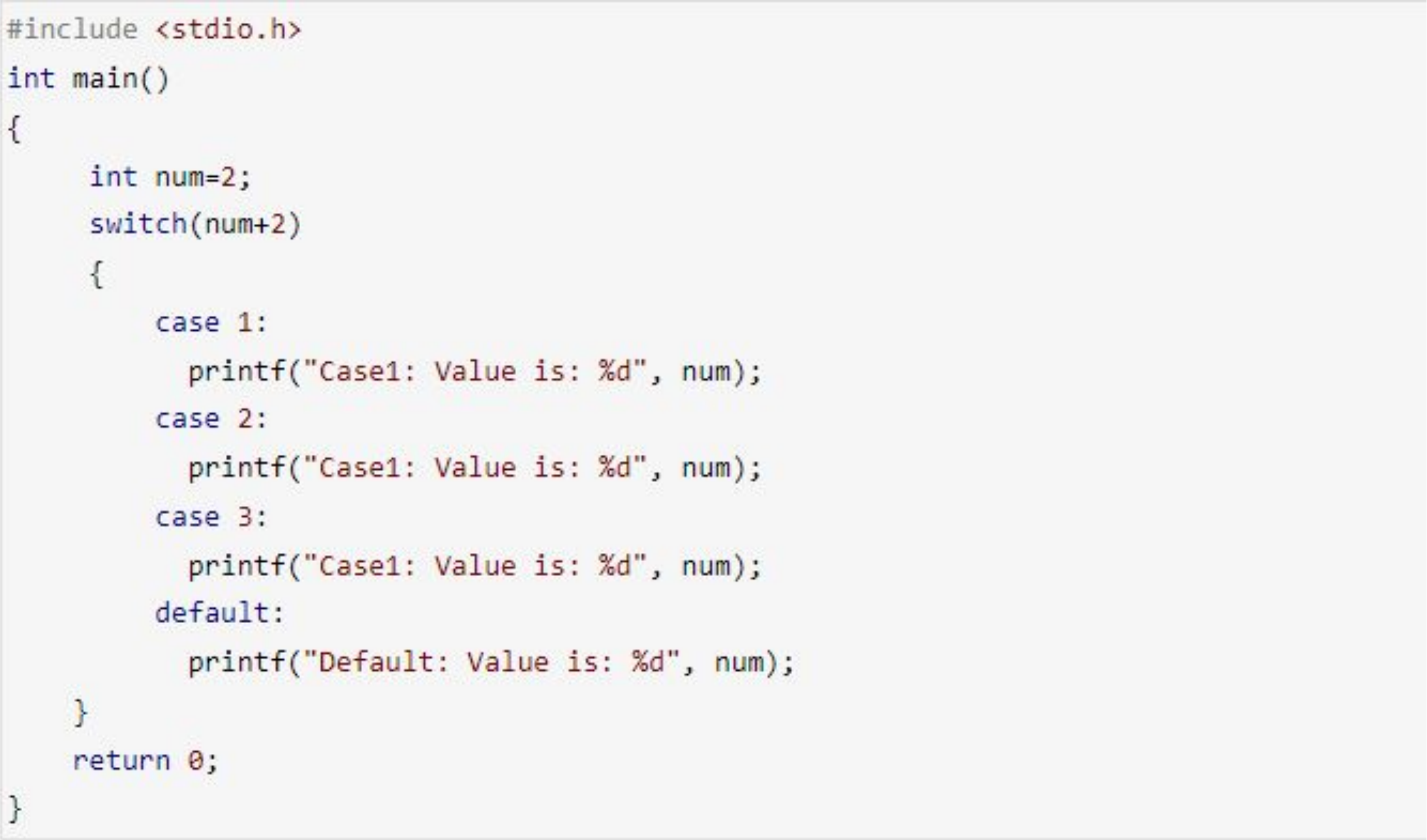
The **switch case statement** is used when we have multiple options and we need to perform a different task for each option.

Switch case statement – Syntax



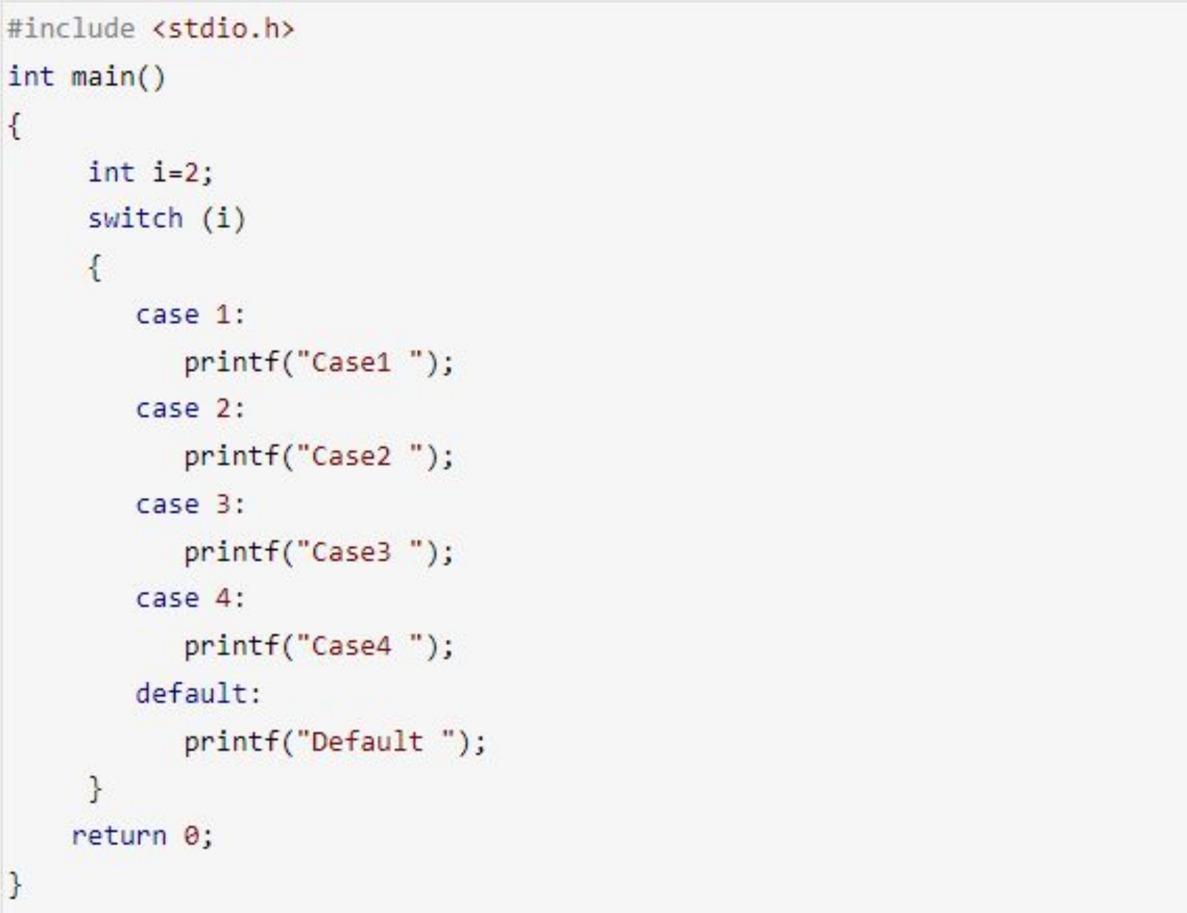
```
switch (variable or an integer expression)
{
     case constant:
     //C Statements
     ;
     case constant:
     //C Statements
     ;
     default:
     //C Statements
     ;
}
```

Switch case statement – Example

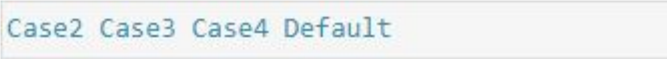


```c
#include <stdio.h>
int main()
{
     int num=2;
     switch(num+2)
     {
         case 1:
           printf("Case1: Value is: %d", num);
         case 2:
           printf("Case1: Value is: %d", num);
         case 3:
           printf("Case1: Value is: %d", num);
         default:
           printf("Default: Value is: %d", num);
    }
    return 0;
}
```

Switch case statement –Break statement

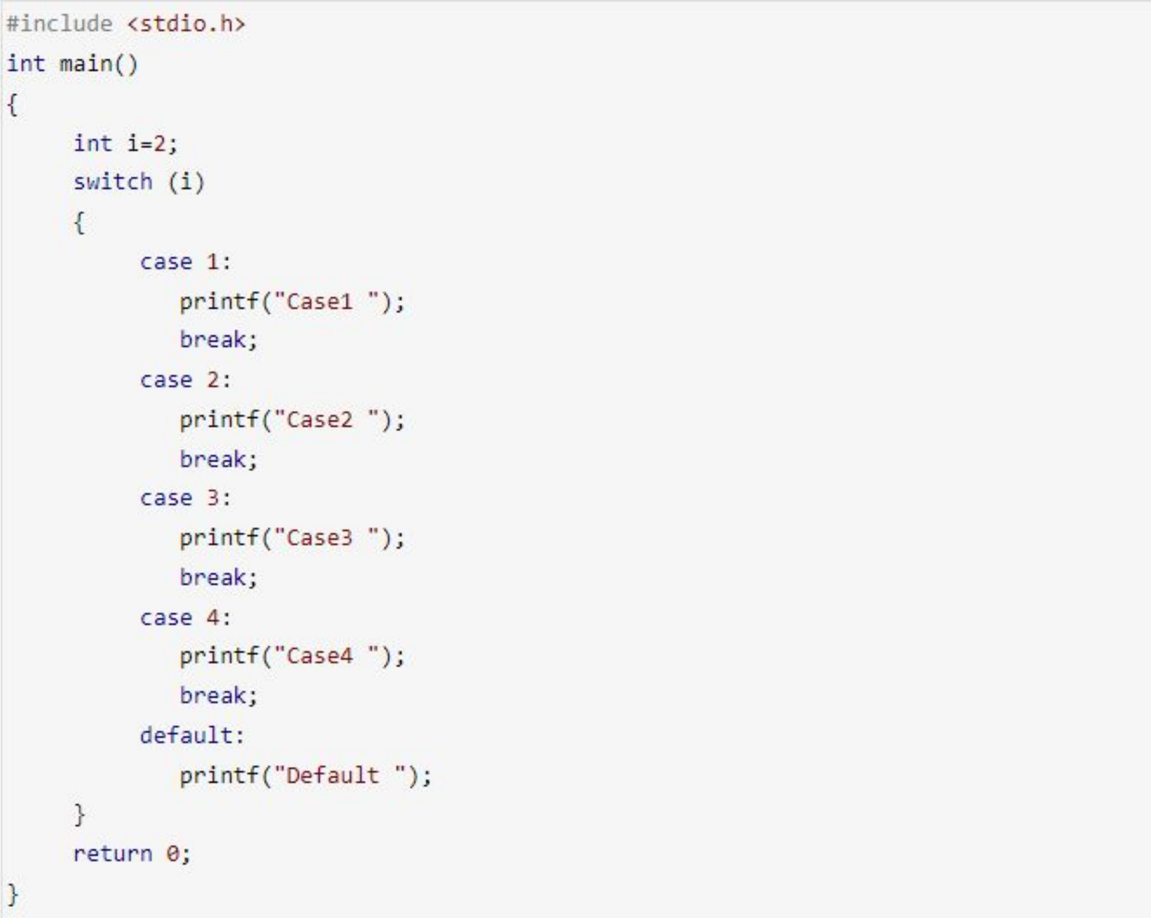


```c
#include <stdio.h>
int main()
{
     int i=2;
     switch (i)
     {
        case 1:
           printf("Case1 ");
        case 2:
           printf("Case2 ");
        case 3:
           printf("Case3 ");
        case 4:
           printf("Case4 ");
        default:
           printf("Default ");
     }
    return 0;
}
```



```
Case2 Case3 Case4 Default
```

Solution



```
#include <stdio.h>
int main()
{
     int i=2;
     switch (i)
     {
          case 1:
             printf("Case1 ");
             break;
          case 2:
             printf("Case2 ");
             break;
          case 3:
             printf("Case3 ");
             break;
          case 4:
             printf("Case4 ");
             break;
          default:
             printf("Default ");
     }
     return 0;
}
```

C++ Loops

The looping can be defined as **repeating the same process** multiple times **until a specific condition satisfies**. There are **three types** of loops used in the C++ language.

Why use loops in C++ language?

The looping simplifies the complex problems into the easy ones. It enables us to alter the flow of the program so that **instead of writing the same code again and again, we can repeat the same code for a finite number of times.** *For example,* if we need to print the first 10 natural numbers then, instead of using the printf statement 10 times, we can print inside a loop which runs up to 10 iterations.

Advantage of loops

1) It provides code reusability.

2) Using loops, we do not need to write the same code again and again.

3) Using loops, we can traverse over the elements of data structures (array or linked lists).

Types of Loops

There are three types of loops:

- for
- While
- do while

For loop

The for loop is used to iterate the statements or a part of the program several times.

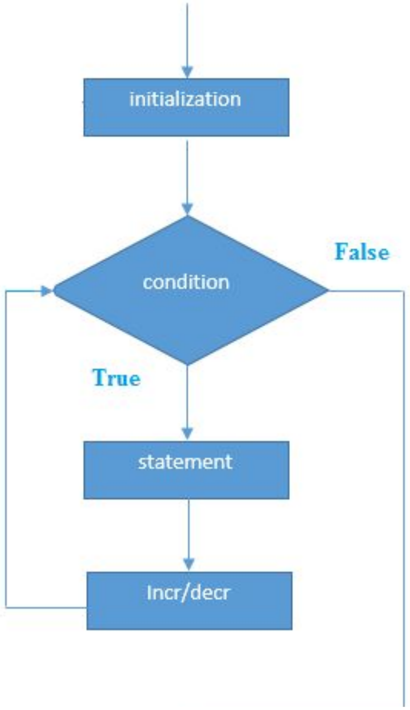
Syntax of for loop

- for(Expression 1; Expression 2; Expression 3){
- //code to be executed
- }

for loop

for loop Examples
Let's see the simple program of for loop that prints table of 1.

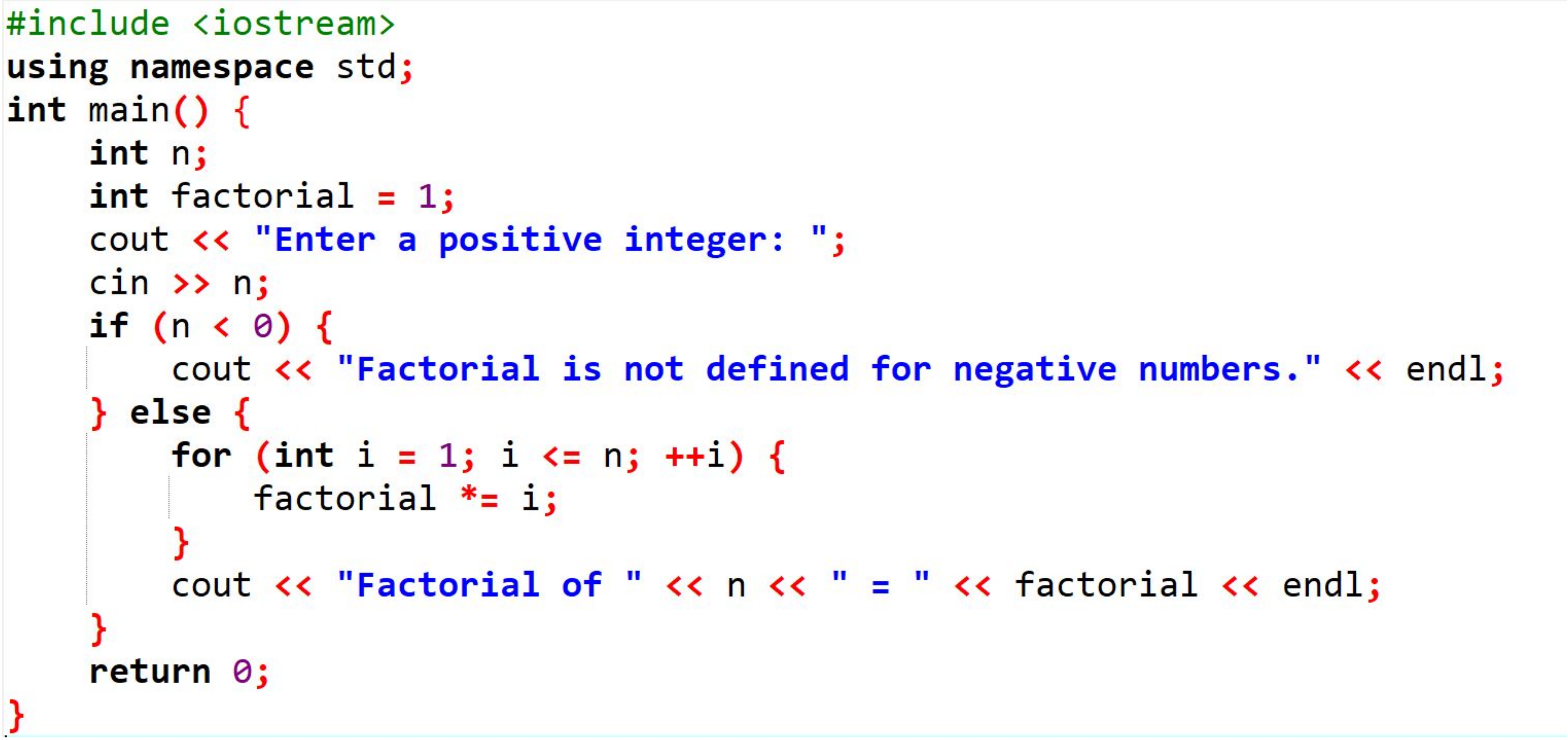
```
#include <iostream>

using namespace std;
int main()
{
int i=0;
for(i=1;i<=10;i++)
{
cout<<i;
}
return 0;
}
```



Practice

Make the program that calculates the factorial of the user given number .

Answer



```
#include <iostream>
using namespace std;
int main() {
    int n;
    int factorial = 1;
    cout << "Enter a positive integer: ";
    cin >> n;
    if (n < 0) {
        cout << "Factorial is not defined for negative numbers." << endl;
    } else {
        for (int i = 1; i <= n; ++i) {
            factorial *= i;
        }
        cout << "Factorial of " << n << " = " << factorial << endl;
    }
    return 0;
}
```

while loop

While loop is also known as a pre-tested loop. In general, a while loop allows a part of the code to be executed multiple times. It can be viewed as a repeating if statement. ***The while loop is mostly used in the case where the number of iterations is not known in advance.***

Syntax of while loop in C language

- while(condition){
- //code to be executed
- }

do-while loop

The do-while loop continues until a given condition satisfies. It is also called post tested loop. **It is used when it is necessary to execute the loop at least once.**

The syntax of do-while loop in c language is given below:

```
do{
//code to be executed
}while(condition);
```

While **vs** do while

While Loop	Do- While Loop
Executes the block of code only **if the condition is true** at the beginning.	Executes the block of code at least once, regardless of the condition, because the condition is checked at the end.
The condition is checked before entering the loop body.	The condition is checked after executing the loop body.
The loop may not execute at all if the condition is false initially.	The loop will execute at least once, even if the condition is false.

pre-tested loop, the condition is checked before the loop body executes.
If the condition evaluates to false initially, the loop body will not execute even once.
post-tested loop, the condition is checked after the loop body executes.
This guarantees that the loop body will always execute at least once, even if the condition is false.

Nested loops

C++ supports nesting of loops. Nesting of loops is the feature that allows the looping of statements inside another loop. Let's observe an example of nesting loops.

Any number of loops can be defined inside another loop, i.e., there is no restriction for defining any number of loops. The nesting level can be defined at n times. You can define any type of loop inside another loop; for example, you can define 'while' loop inside a 'for' loop.

Syntax of Nested loop

```
Outer_loop
{
   Inner_loop
  {
       // inner loop statements.
  }
     // outer loop statements.
}
```

Nested Loops

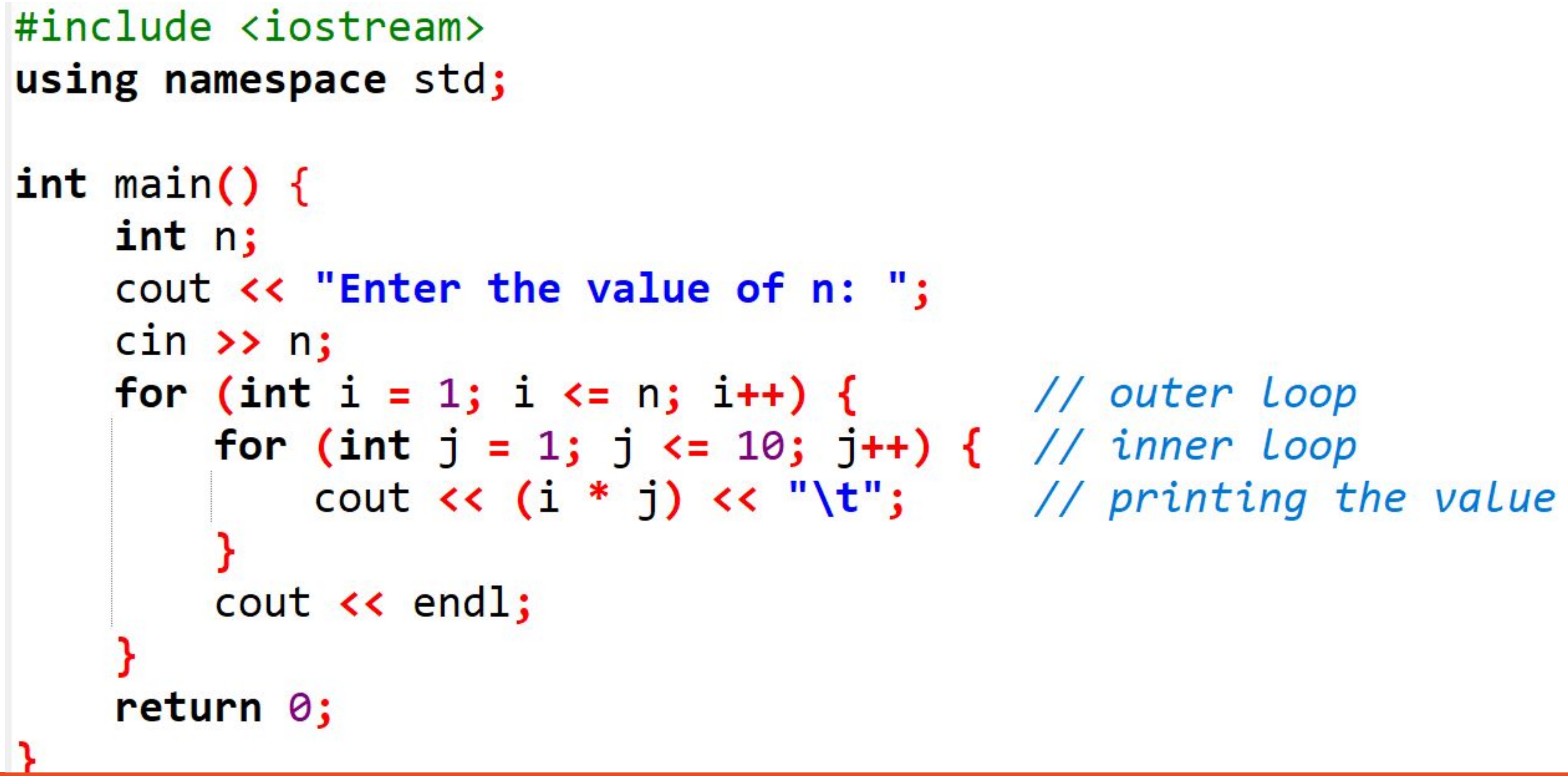
Outer_loop and Inner_loop are the valid loops that can be a 'for' loop, 'while' loop or 'do-while' loop.

Nested for loop

The nested for loop means any type of loop which is defined inside the 'for' loop.

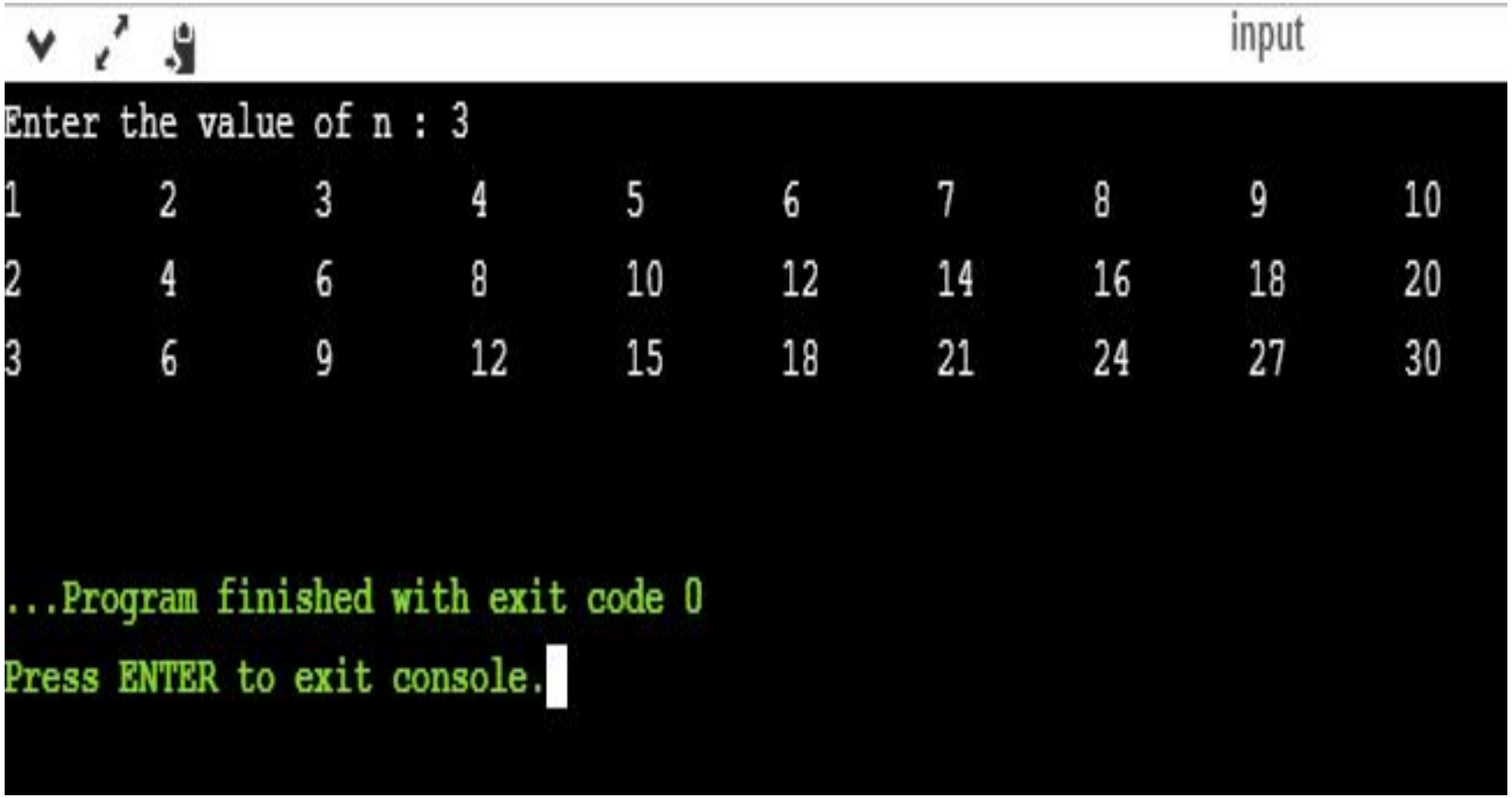
```
for (initialization; condition; update)
{
    for(initialization; condition; update)
    {
          // inner loop statements.
    }
    // outer loop statements.
}
```

Example of nested for loop



```cpp
#include <iostream>
using namespace std;

int main() {
    int n;
    cout << "Enter the value of n: ";
    cin >> n;
    for (int i = 1; i <= n; i++) {       // outer loop
        for (int j = 1; j <= 10; j++) {  // inner loop
            cout << (i * j) << "\t";     // printing the value
        }
        cout << endl;
    }
    return 0;
}
```

Output



```
Enter the value of n : 3
1       2       3       4       5       6       7       8       9       10
2       4       6       8       10      12      14      16      18      20
3       6       9       12      15      18      21      24      27      30


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

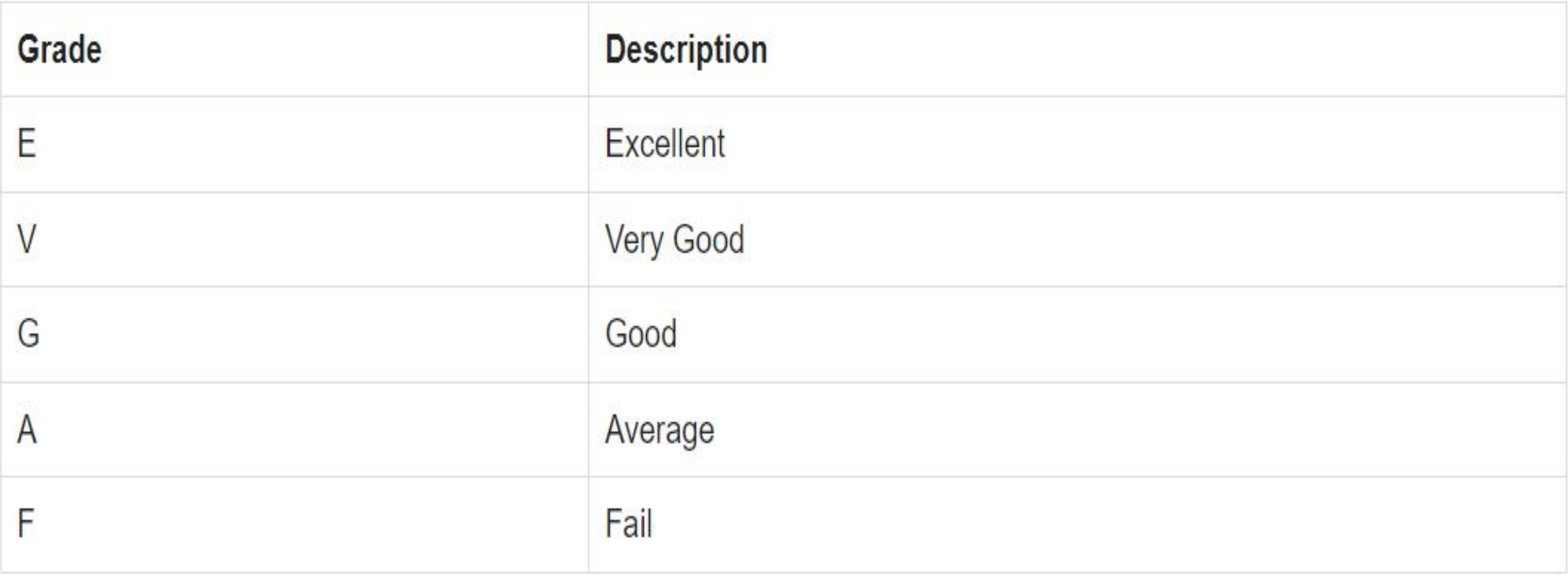
Task-1

Write a C++ program to input electricity unit charge and calculate the total electricity bill according to the given condition:

- For first 50 units Rs. 0.50/unit
- For next 100 units Rs. 0.75/unit
- For next 100 units Rs. 1.20/unit
- For unit above 250 Rs. 1.50/unit
- An additional surcharge of 20% is added to the bill.

Task-2

Write a program in C to accept a grade and declare the equivalent description using switch-case statement :



Grade	Description
E	Excellent
V	Very Good
G	Good
A	Average
F	Fail

Task-3

Write a program in C++ to read any digit (0-9) and display it in the word using switch-case statement.

Task-4

Write a C++ program to display n natural numbers, n will be given by the user.

Task-5

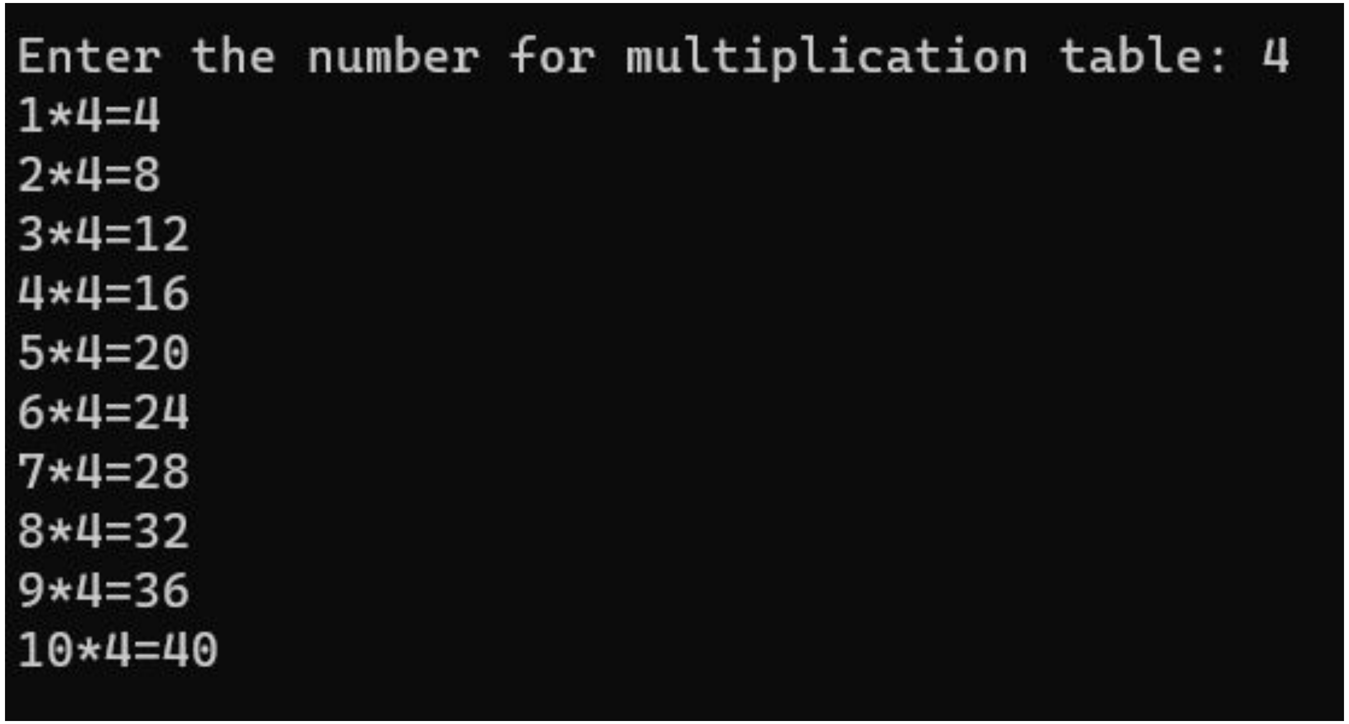
Write a C++ program to find sum of first n even natural numbers.

Task-6

Write down a C++ code to display first n 0dd numbers in reverse order.

Task-7

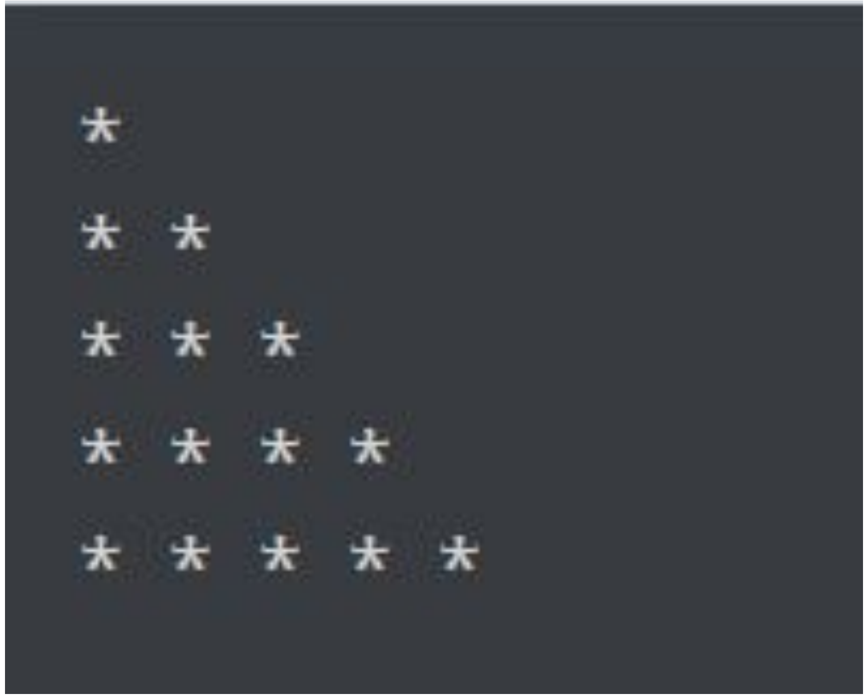
Write down the C++ code to display multiplication table of given



```
Enter the number for multiplication table: 4
1*4=4
2*4=8
3*4=12
4*4=16
5*4=20
6*4=24
7*4=28
8*4=32
9*4=36
10*4=40
```

Task-8

Write down the C++ code to display the following half-pyramid:



```
*
* *
* * *
* * * *
* * * * *
```