Daffodil International University

12 Common Pattern Printing Using C/C++

(Revise the Beginner Sheet 📜)

Written By

Muhammad Anisuzzaman

ID: 191-16-403

Department: Computing and Information System

❖ **<u>Pattern 01:</u> Print Right Triangle Star/Number Pattern**

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

Note: When we solve any Pattern Printing Problem, first we have to think about Rows and Columns such as in each row how many columns we have to print and find the relation between them. Then we can easily solve it. **In this problem,** we see that the row number and the column number are increasing simultaneously. So, the column depends on the row… 🙄

C++ Code: 😃

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

int main(){

    int n = 7;

    for(int row=1; row<=n; row++){

        for(int col=1; col<=row; col++){
            cout << "*" << " ";
        }
        cout << endl; // newline
    }
    return 0;
}
```

If you print **column** value instead of **"*"**, it will print…

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
```

If you print **row** value instead of **"*"**, it will print…

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
6 6 6 6 6 6
7 7 7 7 7 7 7
```

- If you print **col%2** value instead of **“*”**, it will print…

```
1
1 0
1 0 1
1 0 1 0
1 0 1 0 1
1 0 1 0 1 0
1 0 1 0 1 0 1
```

- If you print **row%2** value instead of **“*”**, it will print…

```
1
0 0
1 1 1
0 0 0 0
1 1 1 1 1
0 0 0 0 0 0
1 1 1 1 1 1 1
```

- **Think about this pattern How to print? 😋**

```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
0 1 0 1 0 1
1 0 1 0 1 0 1
```

C++ Code: 😮

```cpp
for(int row=1; row<=n; row++){

    for(int col=1; col<=row; col++){
        ((row+col) % 2 == 0) ?
        cout << "1 ":
        cout << "0 ";
    }
    cout << endl; // newline
}
return 0;
```

- **Note:** I use **ternary operator** because it is handy but you can use **if-else** instead of that…

- Think about **Floyd's Triangle** How to print? 🙄

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27 28
```

C++ Code: 😁

```cpp
int n = 7, count=1;

for(int row=1; row<=n; row++){

    for(int col=1; col<=row; col++){
        cout << count++ << " ";
    }
    cout << endl; // newline
}
return 0;
```

- If you print **(char)(col+64)** value instead of **"*"**, it will print…

```
A
A B
A B C
A B C D
A B C D E
A B C D E F
A B C D E F G
```

- If you print **(char)(row+64)** value instead of **"*"**, it will print…

```
A
B B
C C C
D D D D
E E E E E
F F F F F F
G G G G G G G
```

- **Note:** So, we see that, when we need **different** values, we print **column** value and when we need the **same** value, we print **row** value.

❖ **Pattern 02:** Print Square Star Pattern

C++ Code: 😑

```
for(int row=1; row<=n; row++){

    for(int col=1; col<=n; col++){
        cout << "* ";
    }
    cout << endl; // newline
}
return 0;
```

❖ **Pattern 03:** Print Hollow Square Star Pattern

C++ Code: 😎

```
for(int row=1; row<=n; row++){

    for(int col=1; col<=n; col++){
        (row == 1 || row == n || col == 1 || col == n) ?
        cout << "* ":
        cout << "  "; // here 2 space
    }
    cout << endl; // newline
}
return 0;
```

❖ **Pattern 04: Print Hollow Triangle Star Pattern**

C++ Code: 😐

```
for(int row=1; row<=n; row++){

    for(int col=1; col<=row; col++){
        (col == 1 || row == n || row == col) ?
        cout << "* ":
        cout << "  "; // here 2 space
    }
    cout << endl; // newline
}
return 0;
```

❖ **Pattern 05: Print Mirror Right Triangle Star Pattern**

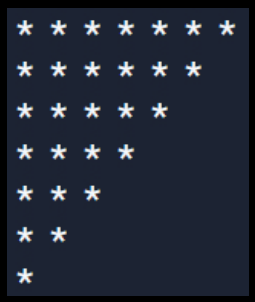
C++ Code: 😀

```
int n = 7;

for(int row=1; row<=n; row++){

    for(int col=1; col<=n-row; col++){
        cout << "  "; // 2 space
    }
    for(int col=1; col<=row; col++){
        cout << "* ";
    }
    cout << endl; // newline
}
return 0;
```

❖ **Pattern 06:** Print Inverted Right Triangle Star Pattern

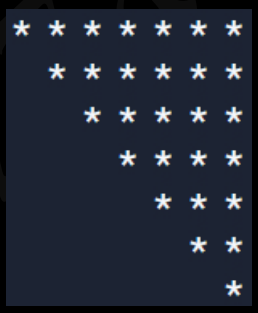


C++ Code: 🙃

```
for(int row=n; row>=1; row--){

    for(int col=1; col<=row; col++){
        cout << "* ";
    }
    cout << endl; // newline
}
return 0;
```

❖ **Pattern 07:** Print Inverted Mirror Right Triangle Star Pattern



C++ Code: 🤪

```
for(int row=n; row>=1; row--){

    for(int col=1; col<=n-row; col++){
        cout << "  ";
    }
    for(int col=1; col<=row; col++){
        cout << "* ";
    }
    cout << endl; // newline
}
return 0;
```

Pattern 08: Print Rhombus Star Pattern

C++ Code: 😕

```
for(int row=1; row<=n; row++){

    for(int col=1; col<=n-row; col++){
        cout << "  ";
    }
    for(int col=1; col<=n; col++){
        cout << "* ";
    }
    cout << endl; // newline
}
return 0;
```

Pattern 09: Print Pyramid Star Pattern

C++ Code: 😳

```
for(int row=1; row<=n; row++){

    for(int col=1; col<=n-row; col++){
        cout << "  "; // 2 space
    }
    for(int col=1; col<=(row*2)-1; col++){
        cout << "* ";
    }
    cout << endl; // newline
}
return 0;
```

Pattern 10: Print Hollow Pyramid Star Pattern

C++ Code: 😶

```
for(int row=1; row<=n; row++){

    for(int col=1; col<=n-row; col++){
        cout << "  "; // 2 space
    }
    for(int col=1; col<=(row*2)-1; col++){
        (col == 1 || row == n || col == (row*2)-1) ?
        cout << "* ":
        cout << "  "; // 2 space
    }
    cout << endl; // newline
}
return 0;
```

Pattern 11: Print Diamond Star Pattern

C++ Code: 😨

```
for(int row=1; row<=n; row++){

    for(int col=1; col<=n-row; col++){
        cout << "  "; // 2 space
    }
    for(int col=1; col<=(row*2)-1; col++){
        cout << "* ";
    }
    cout << endl; // newline
}
for(int row=n-1; row>=1; row--){

    for(int col=1; col<=n-row; col++){
        cout << "  "; // 2 space
    }
    for(int col=1; col<=(row*2)-1; col++){
        cout << "* ";
    }
    cout << endl; // newline
}
return 0;
```

<u>Pattern 12:</u> Print Cross X Star Pattern

C++ Code: 🤢

```
for(int row=1; row<=n; row++){

    for(int col=1; col<=n; col++){
        (row == col || row+col == n+1) ?
        cout << "* ":
        cout << "  "; // 2 space
    }
    cout << endl; // newline
}
return 0;
```