



C++ Course Notes | Fundamentals of Programming - 2 | Week 3

Hello everyone welcome to the weekly lecture notes

Topics to be covered:

1. Break keyword
2. Continue keyword
3. Pattern Printing

Break Keyword

1. Break is a command that allows you to terminate and exit a loop (do while / for / while) or switch commands from any point other than the logical end.
2. It can be used to immediate termination of a loop and works in all the loops (for / while / do while loop).
3. **Syntax :**

```
for( ; ; ) {  
    if(condition) {  
        break;  
    }  
}
```

4. Example:

```
for(int i = 0; i < 10; i++) {  
    if(i == 5) {  
        break;  
    }  
    cout << i << " ";  
}
```

Output:

```
0 1 2 3 4
```

5. NOTE : Break keyword is really useful in case of breaking infinite loops. Example :

```
int i = 0;
while(1) {
    if(i == 5) break;
    cout << i << " ";
    i++;
}
```

Output:

```
0 1 2 3 4
```

Continue keyword

1. The continue keyword is used to end the current iteration in a for loop (or a while loop), and continues to the next iteration.
2. Syntax :

```
for(;;) {
    if(condition) {
        continue;
    }
}
```

3. Syntax (another in case of while loop) :

```
while(condition1) {
    if(condition2) {
        continue;
    }
}
```

4. Example :

```
for(int i = 0; i < 5; i++) {
    if(i == 2) continue;
    cout << i << " ";
}
```

Output:

```
0 1 3 4 5
```

5. Another example (in case of while loop) :

```
int i = 0;
while(i < 5){
    if(i == 2) continue;
    cout << i << " ";
}
```

Output:

```
0 1 3 4 5
```

Pattern Printing

1. **Simple Rectangular Pattern :** In this type of patterns try to find the length and breadth of the rectangle and then form the loop bounds according to that only.

Example:

Pattern :

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

Code:

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

int main() {
    int n = 4, m = 5;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cout << "*";
        }
        cout << '\n';
    }
}
```

Explanation :

Here note the bounds of n and m and print accordingly like in this case n is equal to 4 and m is equal to 5. Don't forget to print the new line when needed.

2. Rectangular Pattern (with spaces) :

Another example of rectangular pattern:

Pattern:

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

Code:

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

int main() {
    int n = 4, m = 5;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (i == 0 || j == 0 || j == m - 1 || i == n - 1) {
                cout << "*";
            } else cout << " ";
        }
        cout << '\n';
    }
}
```

Explanation :

Here see that we even have spaces with the star so in the boundary we print the star and in between we print the spaces. To do this we need to use `if` condition statement here.

3. Rectangular Pattern (using Numbers) :

Pattern :

```
11111
22222
33333
44444
```

Code:

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

int main() {
    int n = 4, m = 5;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cout << (i + 1);
        }
        cout << '\n';
    }
}
```

Explanation:

Here in this case rather than printing the star we print the value of rows and for that we need to just print the value of `i`.

4. Rectangular Pattern (with Numbers and Spaces both)

Pattern:

```
11111
2  2
3  3
44444
```

Code:

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

int main() {
    int n = 4, m = 5;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (i == 0 || i == n - 1 || j == 0 || j == m - 1)
                cout << (i + 1);
            else cout << " ";
        }
        cout << '\n';
    }
}
```

Explanation:

We change the if condition here to the boundary values and only print the row numbers in case of boundary and otherwise we just print spaces.

5. Triangular Pattern:

Pattern :

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

Code:

```
#include <iostream>  
using namespace std;  
  
int main() {  
    int n = 4, m = 5;  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < i + 1; j++) {  
            cout << "*";  
        }  
        cout << '\n';  
    }  
}
```

Explanation:

Here we just loop the inner for loop to only till `i` so that it prints a triangular pattern.

6. Triangular Pattern (with numbers):

Pattern :

```
1  
22  
333  
4444
```

Code:

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

int main() {
    int n = 4, m = 5;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < i + 1; j++) {
            cout << (i + 1);
        }
        cout << '\n';
    }
}
```

Explanation:

Here we just loop the inner for loop to only till `i` so that it prints a triangular pattern and print value of the row instead of print stars.
