# Contents

- C++ Break and Continue
- *for* and *while* loop with “Break” & “Continue”
- C++ functions
- Related example codes

# Break and Continue

You have already seen the `break` statement used in an earlier chapter of this tutorial. It was used to "jump out" of a `switch` statement.

The `break` statement can also be used to jump out of a **loop**.

# Ex-1: *break w/ for loop*

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

int main() {
  for (int i = 0; i < 20; i++) {
    if (i == 5) {
      break;
    }
    cout << i << "\n";
  }
  return 0;
}
```

# Ex-1: *break w/ for loop*

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

int main() {
  for (int i = 0; i < 20; i++) {
    if (i == 5) {
      break;
    }
    cout << i << "\n";
  }
  return 0;
}
```

```
0
1
2
3
4
```

# Ex-2: *continue w/ for loop*

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

int main() {
  for (int i = 0; i < 20; i++) {
    if (i == 5) {
      continue;
    }
    cout << i << "\n";
  }
  return 0;
}
```

# Ex-2: *continue w/ for loop*

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

int main() {
  for (int i = 0; i < 20; i++) {
    if (i == 5) {
      continue;
    }
    cout << i << "\n";
  }
  return 0;
}
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

# Ex-3: *break w/ while loop*

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

int main() {
  int i = 0;
  while (i < 15) {
    cout << i << "\n";
    i++;
    if (i == 5) {
      break;
    }
  }
  return 0;
}
```

```
0
1
2
3
4
```

# Ex-4: *continue w/ while loop*

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

int main() {
  int i = 0;
  while (i < 15) {
    if (i == 5) {
      i++;
      continue;
    }
    cout << i << "\n";
    i++;
  }
  return 0;
}
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
```

# Ex-5: String, array

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

int main() {
  string lectures[4] = {"Physics", "Maths", "History"};
  cout << lectures[0];
  return 0;
}
```

Physics

# Ex-6: Changing values

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

int main() {
  string lectures[4] = {"Physics", "Maths", "History"};
  lectures [1]= ”Chemistry”;
  cout << lectures[1];
  return 0;
}
```

Chemistry

# Ex-7: loop with arrays

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

int main() {
  string lectures[4] = {"Physics", "Maths",
"Chemistry", "History"};
  for(int i = 0; i < 4; i++) {
    cout << lectures[i] << "\n";
  }
  return 0;
}
```

```
Physics
Maths
Chemistry
History
```

# C++'ta Fonksiyonlar

- Fonksiyon, yalnızca çağrıldığında çalışan bir kod bloğudur.
- Parametre olarak bilinen veriler bir fonksiyona aktarılabilir.
- Fonksiyonlar belirli eylemleri gerçekleştirmek için kullanılır ve kodu yeniden kullanmak için önemlidir: Kodu bir kez tanımlanıp, birçok kez kullanılabilir.
- Bir fonksiyon oluşturmak için adını ve ardından parantezlerin () belirtilmesi gerekir.

# C++ Functions

```
void myFunction() {
  // code to be executed
}
```

*syntax*

- myFunction (), is the name of function
- void, there is no value to get back

# Ex-8: C++ function

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

void myFunction() {
  cout << ”it is running!";
}

int main() {
  myFunction();
  return 0;
}
```

İt is running!

# Homework

## Definition of Capacitance

- The **capacitance**, *C*, of a capacitor is defined as the ratio of the magnitude of the charge on either conductor to the potential difference between the conductors

$$C = \frac{Q}{\Delta V}$$

- Capacitance is a positive quantity.
- It is a measure of the ability to store charge
- The SI unit of capacitance is the **farad** (F)
  - large, e.g. microfarads (μF) or picofarads (pF)

## Capacitance

$$C = \frac{Q}{\Delta V} = \frac{Coulombs}{Volts} = Farads$$

*Electric Field between flat plates*

$$E = \frac{\sigma}{\varepsilon_o} = \frac{Q}{\varepsilon_o A}, \quad \Delta V = Ed = \frac{Qd}{\varepsilon_o A}$$

*Potential Difference between flat plates*

$$C = \frac{Q}{\Delta V} = \frac{Q}{Qd/\varepsilon_o A} = \frac{\varepsilon_o A}{d}$$

*Capacitance depends on the Area of the plates and the distance between the plates*