



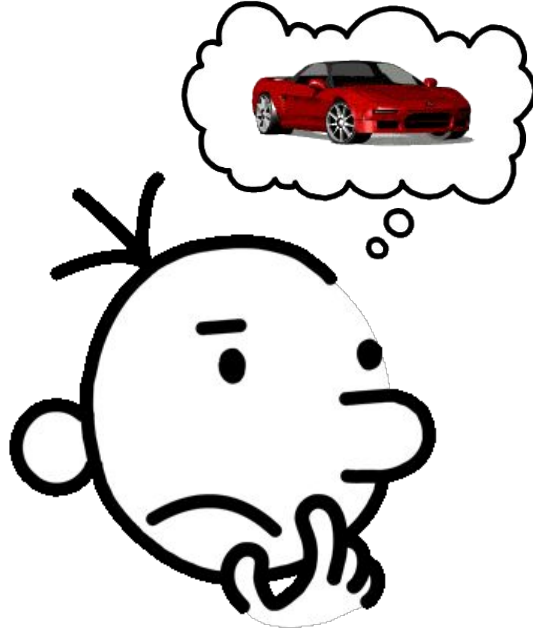
# Conditions in C++



# Why do we need Conditions?

When we add some kind of **condition** on some task, this is called conditional statement.

If I had the **Money**,  
I would buy a **Car**



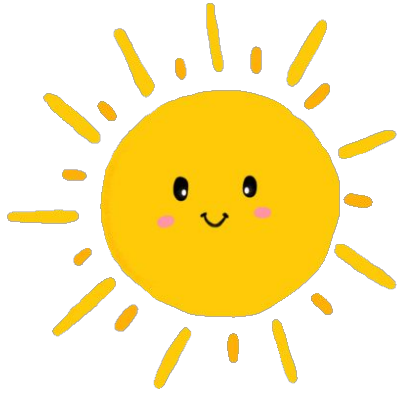
# Why do we need Conditions?

We speak many **Conditional Statements** daily in our life.

If there will be **Good Weather**  
then we will go on **Picnic**.

# Why do we need Conditions?

We speak many **Conditional Statements** daily in our life.



If there will be **no Rain**, then  
we will go to **Market**

*Shopping  
Time* 

# Why do we need **Conditions**?

We speak many **Conditional Statements** daily in our life.



If it will be a **Sunny Day**, then  
we will go to play **Football**.



# Why do we need **Conditions**?

We speak many **Conditional Statements** daily in our life.



If you do my **homework** then i will offer you a **Burger**.



# Why do we need **Conditions**?

As Programming solves **Real World** problems;  
therefore, it also needs the **Conditional Statements**.



# Review: Input/Output in C++

Before understanding **Conditional Statements** in Programming, let's Practice **taking Input** and **displaying Output** which we have already learned in previous week.





# Review: Input/Output in C++

Write a **Program**, that takes **name** as **Input**, and Prints **Welcome with the Name**.



# Review: Input/Output in C++

Write a Program, that takes name as Input, and Prints Welcome with the Name.

```
1  #include <iostream>
2  using namespace std;
3
4  main() {
5      string name;
6      cout << "Please Enter your name: ";
7      cin >> name;
8      cout << "Welcome " << name;
9  }
```

# What should we do?

Now, we want to make changes in the following Program so it prints **Welcome with name** only when the name is **Ahmad**.

```
1  #include <iostream>
2  using namespace std;
3
4  main() {
5      string name;
6      cout << "Please Enter your name: ";
7      cin >> name;
8      cout << "Welcome " << name;
9  }
```

# IF Statement

Now, we want to make changes in the following Program so it prints **Welcome with name** only when the name is **Ahmad**.

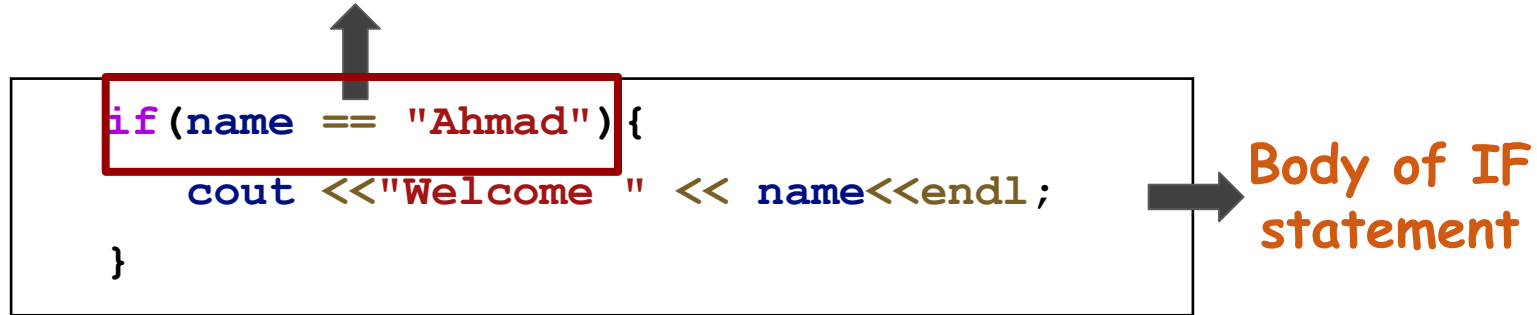
```
1  #include <iostream>
2  using namespace std;
3  main() {
4      string name;
5      cout << "Please Enter your name ";
6      cin >> name;
7      if(name == "Ahmad"){
8          cout << "Welcome " << name << endl;
9      }
10     cout << "Program Ends" << endl;
11 }
```

# IF Statement

This is a conditional statement. **IF** statement has following two parts

- IF statement
- Body of IF statement

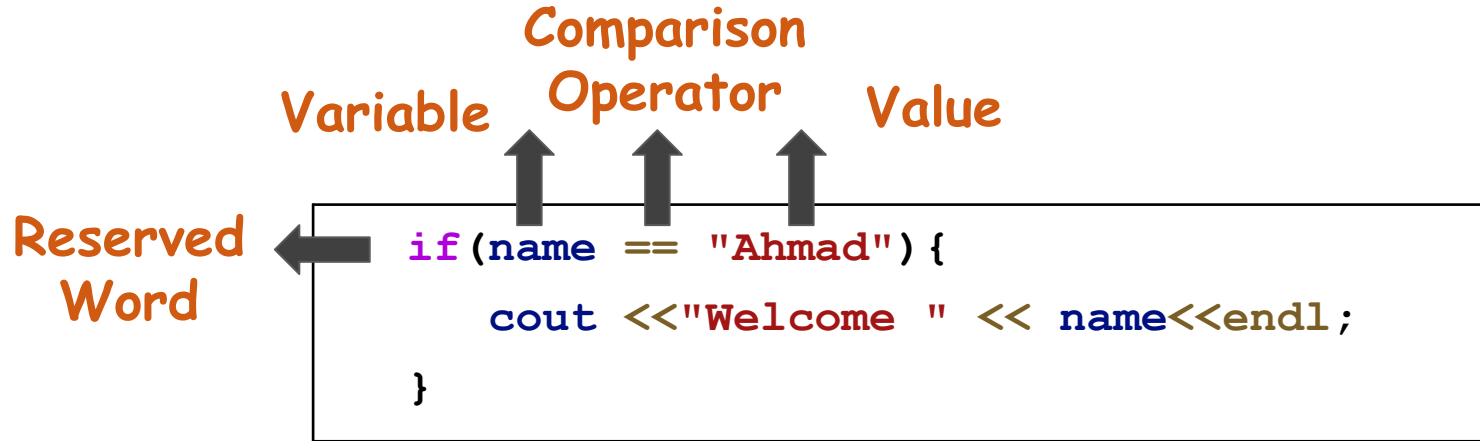
If Statement



# IF Statement

This is a conditional statement. **IF** statement has following two parts

- IF statement
- Body of IF statement



# IF Statement: Boolean Expression

This is a conditional statement. **IF** statement has following two parts

- IF statement
- Body of IF statement

Boolean  
Expression



```
if(name == "Ahmad") {  
    cout <<"Welcome " << name<<endl;  
}
```

# IF Statement: Equal Comparison Operator

Following Program in C++ prints Welcome with name only when the name is Ahmad.

Equal  
(==)  
Comparison  
Operator

```
1  #include <iostream>
2  using namespace std;
3  main() {
4      string name;
5      cout << "Please Enter your name ";
6      cin >> name;
7      if (name == "Ahmad") {
8          cout << "Welcome " << name << endl;
9      }
10     cout << "Program Ends" << endl;
11 }
```



# IF Statement: What Updates?

What if we want to always print **Welcome with name** when the name **is not Ahmad**.

```
1  #include <iostream>
2  using namespace std;
3  main() {
4      string name;
5      cout << "Please Enter your name ";
6      cin >> name;
7      if(name == "Ahmad") {
8          cout << "Welcome " << name << endl;
9      }
10     cout << "Program Ends" << endl;
11 }
```

# IF Statement: Not Equal Comparison Operator

What if we want to always print **Welcome with name** when the name **is not Ahmad**.


Not Equal  
(!=)  
Comparison  
Operator

```
1  #include <iostream>
2  using namespace std;
3  main() {
4      string name;
5      cout << "Please Enter your name ";
6      cin >> name;
7      if (name != "Ahmad") {
8          cout << "Welcome " << name << endl;
9      }
10     cout << "Program Ends" << endl;
11 }
```

# Comparison Operators

Other than Equal (==) and not Equal to (!=) there are many comparison Operators.

# Comparison Operators list



Comparison Operators	Description	Applicable on	Example
==	Equal to	Textual Data Numeric Data	if ( "AB" == "AC" ) if ( 5 == 5 )
!=	Not equal to	Textual Data Numeric Data	if ( "AB" != "AC" ) if ( 5 != 3 )
<	Less than	Numeric Data	if ( 2 > 4 )
>	Greater Than	Numeric Data	if ( 4 < 4 )
<=	Less than or equal to	Numeric Data	if ( 5 <= 90 )
>=	Greater than or equal to	Numeric Data	if ( 66 >= 21 )

# Working Example

Write a program that takes marks of one subject from the user. If marks are more than 50 it displays "You passed" and "Program ends" and if marks are lesser than 50 it displays only "Program ends".



# Expected Outputs

If user enters less than 50

```
Please enter marks : 36  
Program ends
```

If user enters greater than 50

```
Please enter marks : 56  
You passed  
Program ends
```



# Solution

```
1  #include <iostream>
2  using namespace std;
3  main() {
4      int marks;
5      cout<<"Please Enter Marks: ";
6      cin>>marks;
7      if(marks > 50) {
8          cout<<"You are Passed "<<endl;
9      }
10     cout<<"Program Ends";
11 }
```



# Learning Objective

In this lecture, we learnt how to write a **C++ program** for conditional statements with a single **Boolean expression** consisting of a **comparison operator**.





# Conclusion

- To achieve **decision making** in programming, conditional structure is used.
- With the help of comparison operator we can **compare** values.
- A condition is also called a **Boolean expression**.



# Self Assessment

## Solve the Following Programs

1. Write a program that takes the temperature of a patient in Fahrenheit as input and prints "Normal" if the temperature is equal to 98.6

### Test Cases

Input	Output
Temperature Of Patient: 98.6	Normal Program Ends
Temperature Of Patient: 100	Program Ends



# Self Assessment

## Solve Following Programs

2. Write a Program that takes the total price of the items bought by a customer. If the price is exactly equal to 500\$ then it gives an overall 5% discount to the customer and displays the updated price.

### Test Cases

Input	Output
Price: 490	Price after Discount: 490
Price: 500	Price after Discount: 475
Price: 501	Price after Discount: 501



# Self Assessment

## Solve Following Programs

3. Write a program which takes a number as input from the user. If the number is more than 200 it asks the user to enter 4 more numbers and displays the sum of the first 2 numbers and multiplication of the last 2 numbers on separate lines.



# Self Assessment

## Solve Following Programs

4. Write a program which takes a number from the user if the number is not equal to 200, it asks the user to enter 4 more numbers and displays the average of these numbers.

