If it rains, I will stay home → Action 1

otherwise I will come to the class - Action 2

# if –else Statement

If statement is used to implement decision. When a condition is fulfilled, one set of statements is executed. Otherwise, another set of statements is executed.

```
if (Boolean_Expression)
  { statements₁ } → Action 1
else
  { statements₂  } → Action 2
```

**Write a program to check a given grade. If the grade value is greater than or equal to 60, program prints "This student passed" on the console window. Otherwise, program prints "This student failed" on the console window.**

```
#include <iostream>

using namespace std;

int main()
{
    int grade = 52;
    if (grade >= 60)
    {
        cout<<"This student passed ";
    }
    else
    {
        cout<<"This student failed";
    }

    return 0;
}
```

if-else

52 >= 60 → Fail

OUTPUT

This student failed.

**Practice Problem 1. Consider the following if statement to compute the discounted price. What is the discounted price if the original price is 95.**

```
int originalPrice = 95, discountedPrice;

if ( originalPrice > 100 )                 95 > 100  ⟶   False.
       {  discountedPrice = originalPrice -20 ; }
else
       {  discountedPrice = originalPrice -10 ; }
```

= 95 − 10 = 85.

$\boxed{85}$

**Practice Problem 2. Consider the following if statement to compute the discounted price. What is the discounted price if the original price is 100.**

```
                   100
int originalPrice = 95, discountedPrice;

if ( originalPrice > 100 )                100 > 100  ⟶  False.
       {  discountedPrice = originalPrice -20 ; }
else
       {  discountedPrice = originalPrice -10 ; }
```

100 − 10 = 90

$\boxed{90}$

## Comparison Operators

| C++ | Math Notations | Descriptions |
|---|---|---|
| > | > | Greater than |
| >= | ≥ | Greater than or equal |
| < | < | Less than |
| <= | ≤ | Less than or equal |
| == | = | Equal |
| != | ≠ | Not equal |

## Boolean Variables and Operators

Boolean data types have exactly two values: true or false. An expression/condition that returns either true or false is a Boolean expression.

```
bool failed = true;
```

Data-Type

Variable Name

Literals.

condition Decision

“If you fail the exam, you have to retake.” Write a code to implement the following propositional statement.

```
#include <iostream>

using namespace std;

int main()
{
    bool failed = true ;
    if (failed)
    {         true .
        cout<<"You have to retake exam ";

    }
    return 0;
}
```

OUTPUT

```
You have to retake exam .
```