

④ Ternary operator
Shortcut for if-else

Condition ? true-part : false-part ;

if used when there is small conditions

⑤ Switch case

```
Switch (expression) {  
    case 1:  
        // code  
        break;  
    case 2:  
        // code  
        break;  
    default:  
        // code  
}
```

Sample calculator

```
int a, b;  
char op;  
cin >> a >> op >> b;  
switch (op) {  
    case '+':  
        cout << a + b;  
        break;  
    case '-':  
        cout << a - b;  
        break;  
    case '*':  
        cout << a * b;  
        break;  
    case '/':  
        cout << a / b;  
        break;  
    default:  
        cout << "Invalid operator";  
}
```

Exercise: Make a student grade evaluation system.

- Take student name.
- Take marks
- Decide grade.
- Check Pass fail.
- Give special message for distinction

④ Multiple Conditions (and and ||)

and → and

|| → or

⑤ nested if-else

nested means if inside another if

Example

```
int a, b, c;
```

```
cin >> a >> b >> c;
```

```
if (a > b) {
```

```
    if (a > c) {
```

```
        cout << "greater is a";
```

```
    }
```

```
else {
```

```
    cout << "greater is b";
```

```
    }
```

```
}
```

⑥ if-else ladder :

used when you have many conditions

```
if (condition1) {
```

```
    // code
```

```
else if (condition2) {
```

```
    // code
```

```
else if (condition3) {
```

```
    // code
```

```
else {
```

```
    // code
```


char and ASCII

* char stores a single character.

```
char ch = 'a';  
cout << ch; // a
```

* ASCII conversion

```
char ch = 'a';  
cout << (int)ch; // 97
```

```
int z = 65;
```

```
cout << (char)z; // A
```

Chapter-2 (Conditionals)

Conditional means decision making in a program

① if statement

if condition is true code will run otherwise code will

Syntax

```
if (condition) {  
    // code run if condition is true  
}
```

② if-else

if condition is true run if block else run else block

```
if (condition) {  
    // code.  
}
```

```
}
```

```
else {
```

```
    // false
```

```
}
```

③ Comparison operators

= Equal to

!= Not equal

> Greater than

< Less than

>= " equal to

<= " "

Increment / decrement

* Post increment (++)

int x = 4;

cout << x++ << endl; // first print 4

cout << x << endl; // print x = 5

* Pre increment (++)

int x = 4;

cout << ++x << endl; // first print 5

float data type

float = real numbers

float x = 3.1;

float x = 5;

float y = 2;

cout << x/y << endl; // 2.5

Comments in C++

Single line comment

(//) ex: // Koush

Multiline comment

/* this is

line comment */

taking input from a user (std::cin) or (cin)

int n;

cin >> n; → in cin we use double sign.

cout << n; → in cout we use smaller than sign.

type casting (float to int)

int x = 5;

float y = x; // 5.0

float a = 5.9 //

int b = (int)a // 5

always lower value.

C++ (complete Notes)

Chapter (1) - Basics of C++

#

Basic structure of Python.

#include <iostream>

using namespace std;

int main() {

cout << "hello world";

return 0;

}

variables and data types

variable = container (dabba)

```
int x;  
x = 5;
```

```
cout << x << endl; // 5
```

```
x = x + 6;
```

```
cout << x << endl; // 11
```

```
x = 7
```

```
cout << x << endl; // 7
```

endl is use for line change same as \n in C.

Arithmetic operators.

```
int x = 5;
```

```
int y = 2;
```

```
cout << x + y << endl; // 7
```

```
cout << x - y << endl; // 3
```

```
cout << x / y << endl; // 2
```

```
cout << x * y << endl; // 10
```

in C++ int / int give integer ans so, we go 2 in of 2.5 unlike Python.

```
cout << x % y << endl; // 1
```

↳ gives remainder like in Python.

file explains left to right hierarchy (BODMAS X)