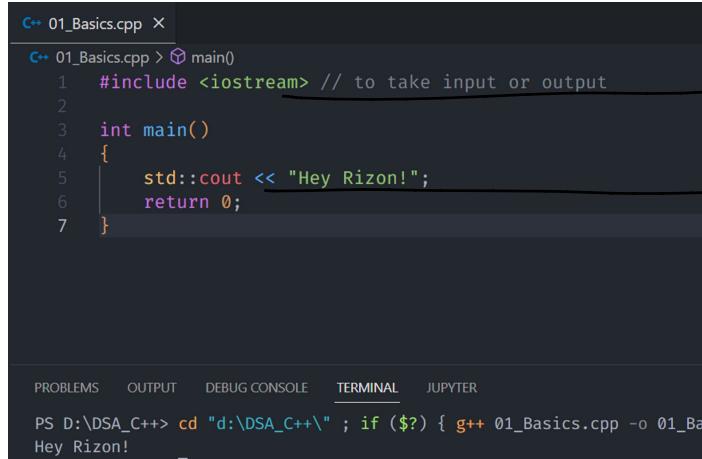


1. C++ Basics



```
01_Basics.cpp >
01_Basics.cpp > #include <iostream> // to take input or output
1 #include <iostream> // to take input or output
2
3 int main()
4 {
5     std::cout << "Hey Rizon!";
6     return 0;
7 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
PS D:\DSA_C++> cd "d:\DSA_C++\" ; if ($?) { g++ 01_Basics.cpp -o 01_Ba
Hey Rizon!
```

→ to take output or input
→ To output we use "cout"

cout << "Hey Rizon" << "ln";
OR
cout << "Hey Rizon" << std::endl;

To avoid std ⇒ we use using namespace std.

To take input: (in)

```
C++ 01_Basics.cpp X
C++ 01_Basics.cpp > main()
1 #include <iostream> // to take input or output
2 using namespace std;
3
4 int main()
5 {
6     int x;
7     cin >> x;
8     cout << "Value of x is : " << x;
9     return 0;
10 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
PS D:\DSA_C++> cd "d:\DSA_C++\" ; if ($?) { g++ 01_Bas
13
Value of x is : 13 ✓
```

To includes all the libraries of C++ it's better to use
#include<bits/stdc++.h> //but it take bit more time

2. Data Types

* int \Rightarrow it's a numeric datatype

for e.g.

```
int x = 10;  
    ↴ variable.  
    ↴ datatype
```

* long long \Rightarrow to store a larger number.

* float \Rightarrow for the decimal.

For e.g. float y = 10.25;

float z = 10;

* double \Rightarrow for the larger decimal value.

* string and getline.

C++ 01_Basics.cpp X

```
5 // int x;
6 // cin >> x;
7 // cout << "Value of x is : " << x;
8 // return 0;
9
10 // Using and getline
11 string s; → string
12 cin >> s;
13 cout << s;
14 return 0;
15 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements.

PS D:\DSA_C++> cd "d:\DSA_C++\" ; if (\$?) { g++ 01_Basics

Hi Rizon
Hi

In string it only picks up anything before the space.

So here for e.g. T/P ⇒ T P Rizon

O(p ⇒ Hi

To take everything in the string then after string use getline.

```
15 // to get whole string use getline
16 string str;
17 getline(cin, str);
18 cout << str;
19 return 0;
20 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

PS D:\DSA_C++> cd "d:\DSA_C++\" ; if (\$?) { g++ 01_B

Hey My Name is Rizon
Hey My Name is Rizon

* character (char) : It will store everything in character.

It has 256 character.

3. If Else

if (condition) {
 // body
}
else {
 // body
}

The screenshot shows a code editor interface with a dark theme. At the top, there are tabs for '01_Basics.cpp', '03_If_else.cpp', and '01_Basics.exe'. The '03_If_else.cpp' tab is active, showing the following code:

```
C++ 03_If_else.cpp > main()
1 #include <iostream>
2 using namespace std;
3
4 // Write a program that takes an input of age
5 // and print if you are adult or notint main()
6 int main()
7 {
8     int age;
9     cin >> age;
10    if (age >= 18)
11    {
12        cout << "You are an adult!";
13    }
14    else if (age < 18)
15    {
16        cout << " You are not an adult!";
17    }
18 }
19 return 0;
```

Below the code editor, there is a terminal window with the following output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
PS D:\DSA_C++> cd "d:\DSA_C++\" ; if ($?) { g++ 03_If_else.cpp }
18
You are an adult!
PS D:\DSA_C++> 
```

Problem Statement

```
23 /*
24 A school has following rules for the grading system
25 a. Below 25 - F
26 b. 25 - 44 - E
27 c. 45 - 49 -D
28 d. 50 - 59 - C
29 e. 60 - 79 - B
30 f. 80 - 100 - A
31 */
```

```
13 int main() {
14     int marks;
15     cin >> marks;
16     if(marks < 25) {
17         cout << "F";
18     }
19     else if(marks <= 44) {
20         cout << "E";
21     }
22     else if(marks <= 49) {
23         cout << "D";
24     }
25     else if(marks <= 59) {
26         cout << "C";
27     }
28     else if(marks <= 79) {
29         cout << "B";
30     }
31     else if(marks <= 100) {
32         cout << "A";
33     }
34     return 0;
35 }
```

```
2 using namespace std;
3 /*
4 Take the age from the user and then decide accordingly
5 1. If age < 18,
6     print-> not eligible for job
7 2. If age >= 18,
8     print-> "eligible for job"
9 3. If age >= 55 and age <= 57,
10    print-> "eligible for job, but retirement soon."
11 4. If age > 57
12    print-> "retirement time"
13 */
14 int main() {
15     int age;
16     cin >> age;
17     if(age < 18) {
18         cout << "not eligible for job";
19     }
20     // >= 18
21     else if(age <= 57) {
22         cout << "eligible for job";
23         if(age >= 55) {
24             cout << ", but retirement soon";
25         }
26     }
27     else {
28         cout << "retirement time";
29     }
}
```

```
27     else {
28         cout << "retirement time";
29     }
30     return 0;
31 }
```

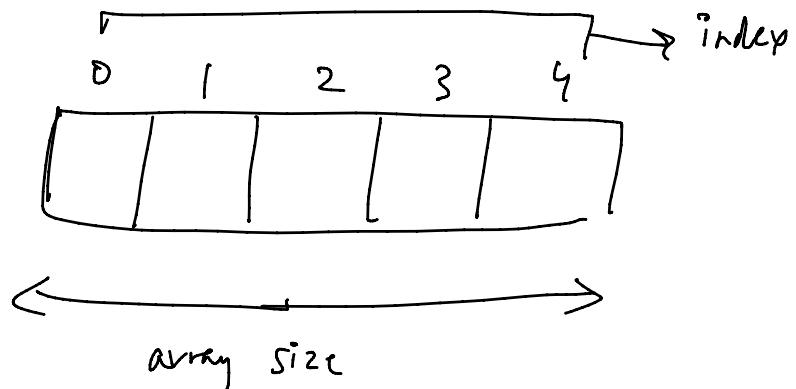
4. Switch

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 /*
4 Take the day no and print the corresponding day
5 for 1 print Monday,
6 for 2 print Tuesday and so on for 7 print Sunday.
7 */
8 int main() {
9
10     return 0;
11 }
```

```
12     switch(day) {
13         case 1:
14             cout << "Monday";
15             break;
16         case 2:
17             cout << "Tuesday";
18             break;
19         case 3:
20             cout << "Wednesday";
21             break;
22         case 4:
23             cout << "Thursday";
24             break;
25         case 5:
26             cout << "Friday";
27             break;
28         case 6:
29             cout << "Saturday";
30             break;
31         case 7:
32             cout << "Sunday";
33             break;
34     }
35     return 0;
36 }
```

5. Array

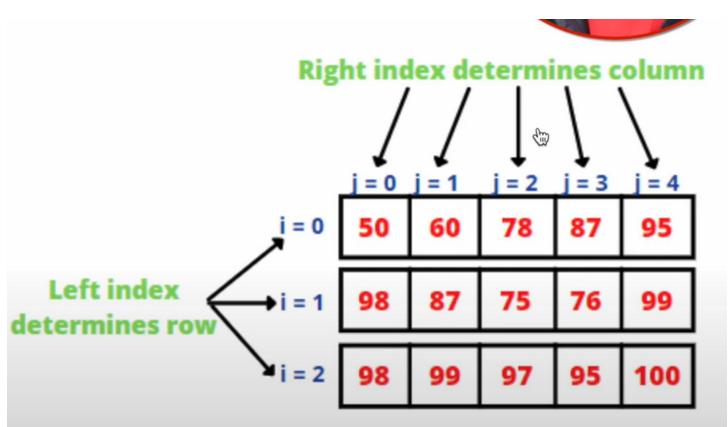
Collections of same data type stored at contiguous memory locations.



```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int arr[5];
6     cin >> arr[0] >> arr[1] >> arr[2] >> arr[3] >> arr[4];
7
8     arr[3] = 16;
9     cout << arr[3];
10
11 }
```

2D array

int arr [3] [5] \Rightarrow
 ↑
row col



6. Strings

String stores every character in terms of index.

String $s = "Rizwan"$

cout $\ll s[2];$ // C

// To find the length of the size

int length = s.size();

// To find the last value of the string

cout $\ll s[length - 1]$

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     string s = "Striver";
6     int len = s.size();
7     cout << s[len - 1];
8     return 0;
9 }
```

7. For Loops

```
for ( initialization; condition; increment/decrement ) {  
    body  
}
```

8. While Loops

While:

initialization;

while (condition) {

 body

 increment / decrement

}

Do while: It's used because it runs atleast one time.

do {

 body

 increment / decrement

}

} while (condition)

9. Functions

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 // Functions are set of code which performs something for you
4 // Functions are used to modularise code
5 // Functions are used to increase readability
6 // Functions are used to use same code multiple times
7 // void -> which does
8 // return
9 // parameterised
10 // non parameterised
11
12 int main() {
13
14     return 0;
15 }
```

```
11
12 void printName() {
13     cout << "hey Striver!";
14 }
15 int main() {
16     printName();
17     return 0;
18 }
```

```
11
12 void printName(string name) {
13     cout << "hey " << name;
14 }
15 int main() {
16     string name;
17     cin >> name;
18     printName(name);
19     return 0;
20 }
```

```
input.txt
1 AMan

output.txt
1 hey AMan
```

Pass by Value : It took copy , so final value of it doesn't change.

```
12
13 // pass by value
14 void doSomething(int num) {
15     cout << num << endl;
16     num += 5;
17     cout << num << endl;
18     num += 5;
19     cout << num << endl;
20 }
21 int main() {
22     int num = 10;
23     doSomething(num);
24     cout << num << endl;
25     return 0;
26 }
```

Value didn't change.

```
12
13 // pass by value
14 void doSomething(string s) {
15     s[0] = 't';
16     cout << s << endl;
17 }
18 int main() {
19     string s = "raj";
20     doSomething(s);
21     cout << s << endl;
22     return 0;
23 }
```

Pass by Reference \Rightarrow Original Value get changes.

```
13 // pass by reference
14 void doSomething(string &s) {
15     s[0] = 't';
16     cout << s << endl;
17 }
18 int main() {
19     string s = "raj";
20     doSomething(s);
21     cout << s << endl;
22     return 0;
23 }
```

Array should always pass by reference.