

Computer programming

Section 2



Agenda

- Our first Program.
- Basic operations
- What's a library and how to use it?
- Branching. Write a program with multi execution paths.

Our first program

```
1 | int main(){  
2 | .....  
3 | ..... }
```

- Any program should contain a “**main**”
- Curly brackets determine the starting and ending of block of code.
- What does our first program do?!

Basic operations

- What does a computer can do?!

```
1  int main(){
2  | 3+6;
3  }
```

E:/CodeLite/test/yyy/main.cpp:2:7: warning: statement has no effect [-Wunused-value]

Storing data

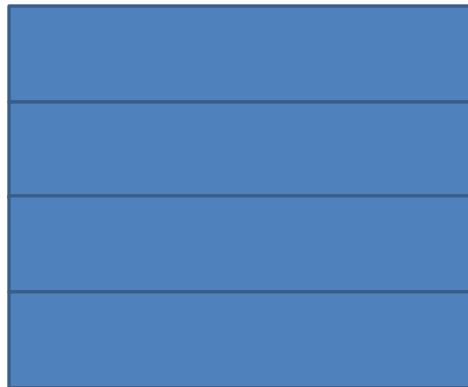
- To Store data on computer's memory according to C++ language you will need to **define** the following parameters:
 - Data type
 - Identifier(Name)
 - Scope.

Storing data cont.

```
1 | int main(){  
2 |     int sum;  
3 |     ...  
4 | }
```

Memory

sum



- This line define a variable with data type **int (integer)** with the **identifier** “sum”
- What’s the scope?

Example

```
1  |  ▾ int main(){
2  |  ▾  ... {
3  |  |  ▸  ... int sum;
4  |  |  ... }
5  |  |  ▸  ... sum = 3+6;
6  |  |  ... }
-  |
```

E:/CodeLite/test/yey/main.cpp:3:13: warning: unused variable 'sum' [-Wunused-variable]

E:/CodeLite/test/yey/main.cpp:5:5: error: 'sum' was not declared in this scope

Basic Data type in C++ language

- **char**
- **int**
- **float**
- **double**
- **void**

Every data type requires a certain memory size to be allocated in

Valid Variable Identifier

- Don't use a keyword (i.e if, else, ...)
- Don't use spaces.
- Don't use symbols (i.e -, &, %, ...)
- Start the name with letter or underscore.
- Note: C++ is case sensitive language.

- Valid Identifiers: Ahmed1, _user2, sum_3
- Invalid Identifiers: 1Num, int, Sheet 1, Lec-1

Basic operators

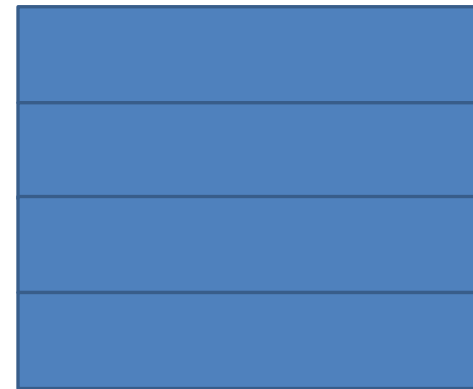
- Assignment operator
- Arithmetic operator
- Relational operator
- Logic operator

Assignment operator

```
1 | ▾ int main(){  
2 |   int sum;  
3 |   ...  
4 |   }
```

sum

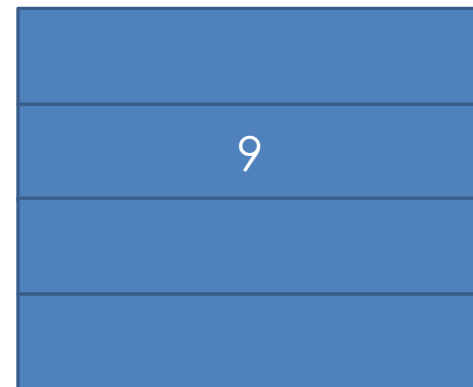
Memory



```
1 | ▾ int main(){  
2 |   int sum;  
3 |   sum = 3+6;  
4 |   }
```

sum

Memory



Arithmetic operators

Operator	Action
-	Subtraction, also unary minus
+	Addition
*	Multiplication
/	Division
%	Modulus

- Evaluation of mathematical expression
 - $3+4*2 = ?$
- These operators have an execution arrange that you should take care of.
- Use brackets to be in safe side.
- $(3+4)*2$ or $3+(4*2)$

Example

```
1  ▾ int main(){
2  ... int num1;
3  ... int num2 = 4;
4  ... num1 = 10;
5  ▶ ... int sum;
6  ▶ ... int subtraction;
7  ▶ ... int multiply;
8  ...
9  ... sum = num1 + num2;
10 ... subtraction = num1-num2;
11 ... multiply = num1*num2;
12 ... }
```

What's a library?

- We need to do further more operations in c++ language as taking an input from the user or displaying output.
- To extend our c++ language capabilities we can use c++ language standard library.
- A library is a set of functions that are previously saved.

Using c++ standard library

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int sum;
6     ...
7 }
```

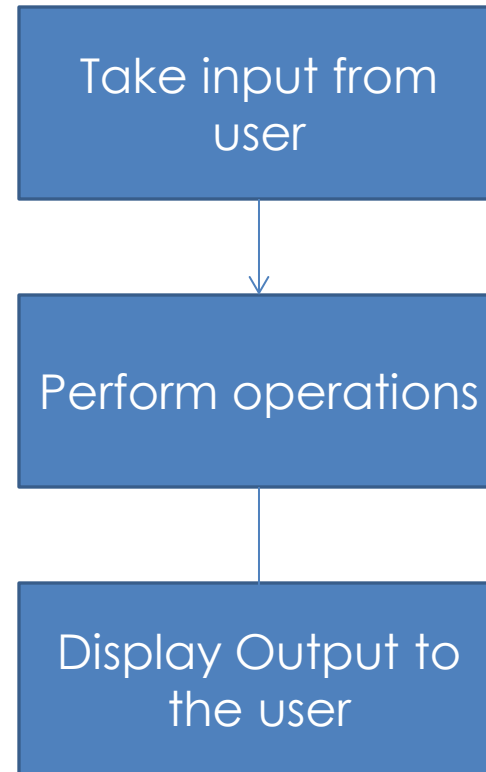
Most common functions in the standard library

- `cout<<`
- `cin>>`

- Example

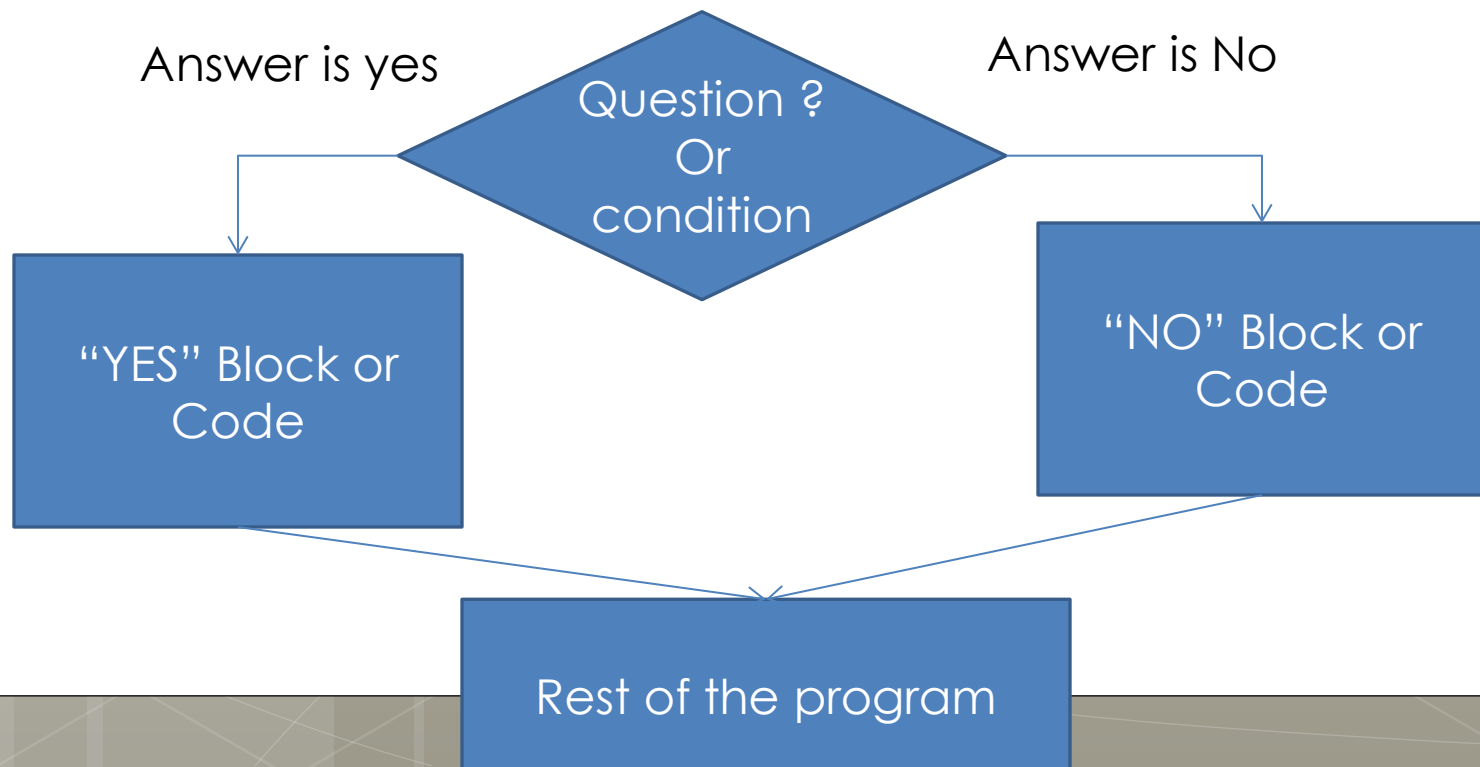
What program can I do till this moment?

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int num1;
6      int num2;
7      int division;
8      cout << "Please enter the first number" <<endl;
9      cin>>num1;
10     cout << "Please enter the second number" <<endl;
11     cin>>num2;
12     division = num1/num2;
13     cout <<"The division is " << division <<endl;
14 }
```



Branching statements

- More complex programs usually have multi execution paths.



If statement

```
if(Condition)
```

```
{
```

```
//Do this block of code if condition
```

```
//evaluation equals 1
```

```
}
```

```
else
```

```
{
```

```
//Do this block of code if condition
```

```
//evaluation equals 0
```

```
}
```

If statement

```
if(1)
{
//Do this block of code if condition
//evaluation equals 1
}
```

```
else
{
//Do this block of code if condition
//evaluation equals 0
}
```

Skipped

If statement

```
if(0)
```

```
{
```

```
//Do this block of code if condition
```

```
//evaluation equals 1
```

```
}
```

```
else
```

```
{
```

```
//Do this block of code if condition
```

```
//evaluation equals 0
```

```
}
```

Skipped

Relational operators

Operator

>

>=

<

<=

==

!=

Action

Greater than

Greater than or equal

Less than

Less than or equal

Equal

Not equal

Logic operators

Operator

&&

||

!

Action

AND

OR

NOT

Switch case

```
int index;  
switch(index){  
    case 1:  
        //Do something  
        break;  
    case 2:  
        //Do Another thing  
        break;  
    default|  
        //If the input is not in the cases do this  
}
```




What program can we do till
the moment?

- Let's make a “**MAC Cashier**” :D

Assignment

- You should do a calculator