DATA STRUCTURE

Introduction to C++

Collected and prepared by

Mahmoud Khaled Gamal Elfouly

mahmoud.elfouly@su.edu.eg mahmoud.khaled21699@gmail.com

What is a C++?

- ✓ Object-oriented
- ✓ Simple
- ✓ General-purpose
- ✓ Compiled

Set Up the Programming Environment

There are many free IDEs (Integrated Development Environments) for C++, such as

Codeblocks

https://www.codeblocks.org/downloads/binaries/

(codeblocks-20.03mingw-setup.exe)

Dev-C++

https://orwelldevcpp.blogspot.com/

Visual Studio 2022

https://visualstudio.microsoft.com/

The Parts of a C++ Program

Program 1

```
//A simple C++ program
#include<iostream>
using namespace std;
```

```

int main() {
  cout << "Hello World";
  return 0;
}

```

Output

```
Hello World
```

`#include`

It means "copy here."

`<iostream>`

It is a header file library that lets us work with input and output objects, such as `cout`

`Using namespace std;`

The using command tells the compiler to allow all the names in the “`std`” namespace to be usable without their prefix.

Note

Without using the std namespace, the names would have to include the prefix and be written as

`std::cout, std::cin, and std::endl.`

`int main () {}`

This is called a **function**. Any code inside its curly brackets `{}` will be executed.

`return 0`

Ends the main function. It denotes that the application run without error.

Every C++ statement ends with a semicolon (;)

C++ Output

The cout object, together with the insertion operator <<, is used to output values/print text.

Example

```
cout << "Hello World" << endl; //Output => Hello World
```

Code	Description	Example

\n or endl	To make a new line	`cout << "Hello World" << endl;`
		`//Output => Hello World`
		`cout << "Hello World \n";`
		`//Output => Hello World`

\t	To create a horizontal tab	`cout << "Hello \t World";`
		`//Output => Hello    World`

\\	To insert a backslash character (\).	`cout << "Hello \\ World";`
		`//Output => Hello \ World`

\”	To insert a double quote character.	`cout << "Hello \"World\" ";`
		`//Output => Hello "World"`

\’	To insert a single quote character.	`cout << "Hello \'World\' ";`
		`//Output => Hello 'World'`

Data Types

The data type specifies the size and type of information the variable will store. (Primitive Data Types)

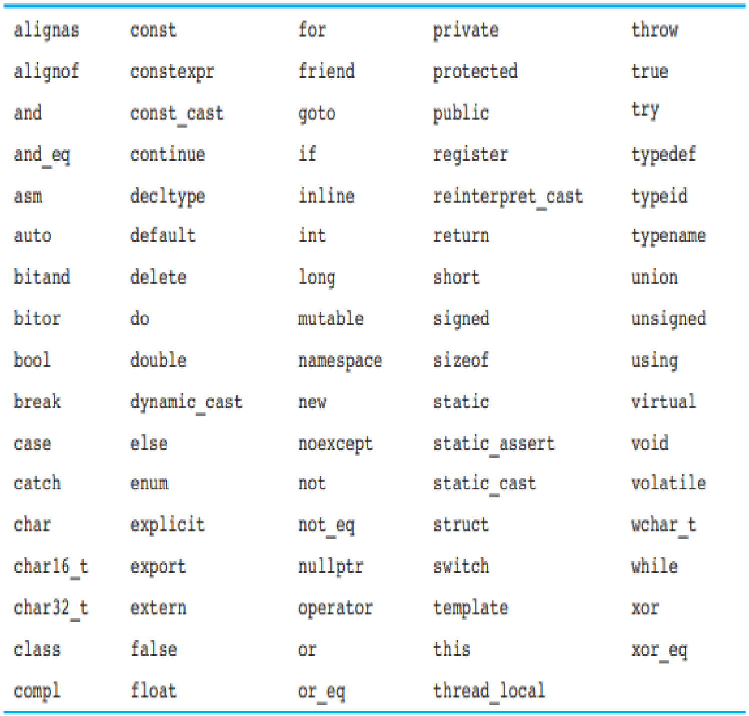
Data Type	Size (bytes)	Example
Int	4	`int size = 5;`
long long	8	`long long size;`
Float	4	`float number = 5.5;`
Double	8	`double length = 55.5;`
Bool	1	`bool state = true;`
Char	1	`char letter = 'A';`

Variables

Variables are containers for storing data values.

The General Rules	Examples
Names can contain letters, digits, and underscores.	✓ X ✓ First_name ✓ Counter1
Names must begin with a letter or an underscore (_)	✓ _Name ✓ Check
Names are case sensitive.	✓ FirstName ✓ Firstname

	Two variables are different
Names cannot contain whitespaces or special characters like !, #, %, etc.	✗ First name ✗ !yes ✗ Name#
C++ keywords cannot be used as names.	✗ Int ✗ double ✗ bool



alignas
alignof
and
and_eq
asm
auto
bitand
bitor
bool
break
case
catch
char
char16_t
char32_t
class
compl

const
constexpr
const_cast
continue
decltype
default
delete
do
double
dynamic_cast
else
enum
explicit
export
extern
false
float

for
friend
goto
if
inline
int
long
mutable
namespace
new
noexcept
not
not_eq
nullptr
operator
or
or_eq

private
protected
public
register
reinterpret_cast
return
short
signed
sizeof
static
static_assert
static_cast
struct
switch
template
this
thread_local

throw
true
try
typedef
typeid
typename
union
unsigned
using
virtual
void
volatile
wchar_t
while
xor
xor_eq

Tip 1

Choose variable names that indicate **what the variables are used for**

Declare Variables	Declare and initialize variables
`int x;`	`int x=5;`
`float y, z;`	`float y=4.2, z=6.2;`
`bool check;`	`bool check= true;`
`double length;`	`double length=5.2;`

Comments

Comments are **notes of explanation that document lines or sections of a program**.

Comments are **part of the program, but the compiler ignores them**.

They are **intended for people who may be reading the source code**.

Single-line comments

```
      // This is a comment
```

Multi-line comments

```
      /*
            Multi-line
            comments
      */
```

C++ Input User

The `cin` object

It can be used to read data typed at the keyboard.

```
int y;
```

```
cout << "Plz, Enter a number of y ";
cin >> y;
cout << "the value of y is " << y << endl;
```

Entering Multiple Values

```
int length, width;

cin >> length >> width;
```

Arithmetic Operators

Operator	Example
+	x + y
-	x – y
*	x * y
/	x / y
%	x % y
++	x++ , ++x
--	x-- , --x

Assignment Operators

Operator	Example	Same as
=	x=5	x=5
+=	x+=5	x=x+5
-=	x-=5	x=x-5
*=	x*=5	x=x*5
/=	x/=5	x=x/5
%=	x%=5	x=x%5

Decision Making

If, else if, else

It can evaluate any condition or expression that results in a Boolean value.

```
if (condition1) {

  // block of code to be executed if condition1 is true

} else if (condition2) {

  // block of code to be executed if the condition1 is false and condition2 is true

} else {

  // block of code to be executed if the condition1 is false and condition2 is false

}
```

Inline if statement

```
variable = (condition) ? expressionTrue : expressionFalse;
```

Switch Case

switch works with integer types, characters, and enums.

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

Comparison Operators

Operator	Example
==	x==5
!=	x!=5
>	x>5
<	x<5
>=	x>=5
<=	x<=5

Logical Operators

Operator	Example
&& (and)	(x == 5 && y==2)
|| (or)	(x == 3 || y==3)
! (Not)	!(x<5)

For Loop

Syntax

```
for ( initialization; test; update)
{
       statement;
       statement;
       // Place as many statements here
}
```

`int i =1;`

initialization

`i<=10`

condition

Example

```
for(int i=1;i<=10;i++){

    cout<<i<<" ";

}
```

`i++`

Increment i by 1

Output

```
1 2 3 4 5 6 7 8 9 10
```

While Loop

Syntax

```
while ( expression) {
         statement;
         statement;
         // Place as many statements here
 }
```

Example

```
int i = 0;

while (i < 10) {

  cout << i << " ";

  i+=2;

}
```

int i =0;

initialization

i < 10

condition

i+=2

Increment i by 2

Output

```
0 2 4 6 8
```

Do … While Loop

Syntax

```
do {
       statement;
       statement;
        // Place as many statements here
 } while ( expression);
```

Example

```
int i = 4;

do {

  cout << i << " ";

  i--;

} while (i >= 0);
```

```
int i = 4;
```
initialization

```
i--;
```
decrement i by 1

```
i >= 0
```
condition

Output

```
4 3 2 1 0
```

Predict Output

```
1- int x = 12;
   cout << -- x + 1<<", ";
   cout<< x++ <<", ";
   cout<<x;

2- int grade = 100;
   if (grade > 60 && grade < 100)
      cout << "Pass";
   else cout << "Fail";

3- int x = 1, y = 2;
   cout << x - y + 3 * 4 / 5;

4- int x = 1, y = 2;
   switch(y)
   {
       case 1: cout << "one";
       case 2: cout << "two";
       default: cout << "three";
   }

5- double a = 2.5;
   int b = 3;
   b = a;
   cout << b;

6- int x = 5, y = 7;
   if (! (x< y))
     cout << "x does not equal y";
```

```
else
  cout << "x equals y";
```

```
7- int x=2,y=3,z=4;
   y+=x++;
   z-=--y;
   cout<<x<<","<<y<<","<<z;
```

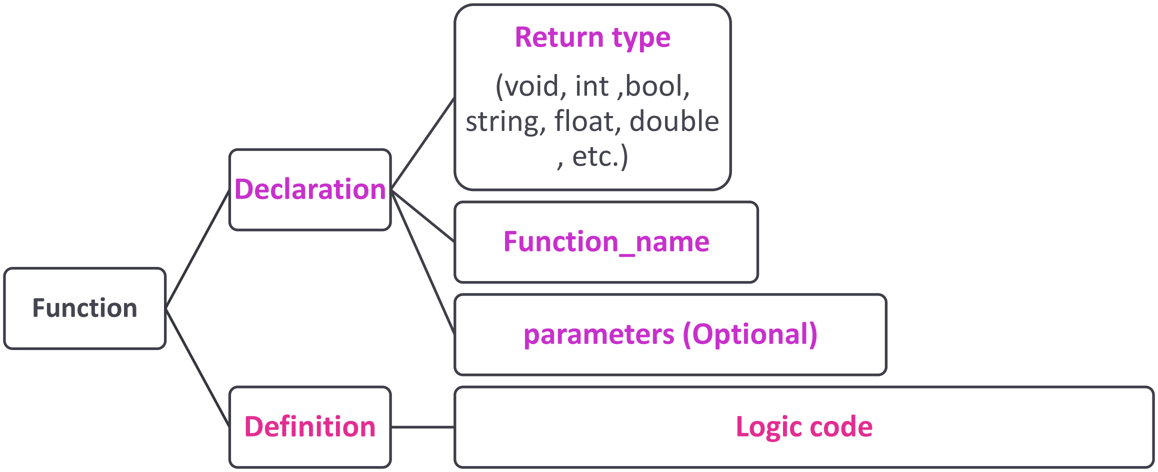
```
8- int x=2,y=3;
   cout<<x * y % 4 + 11 / x - y;
```

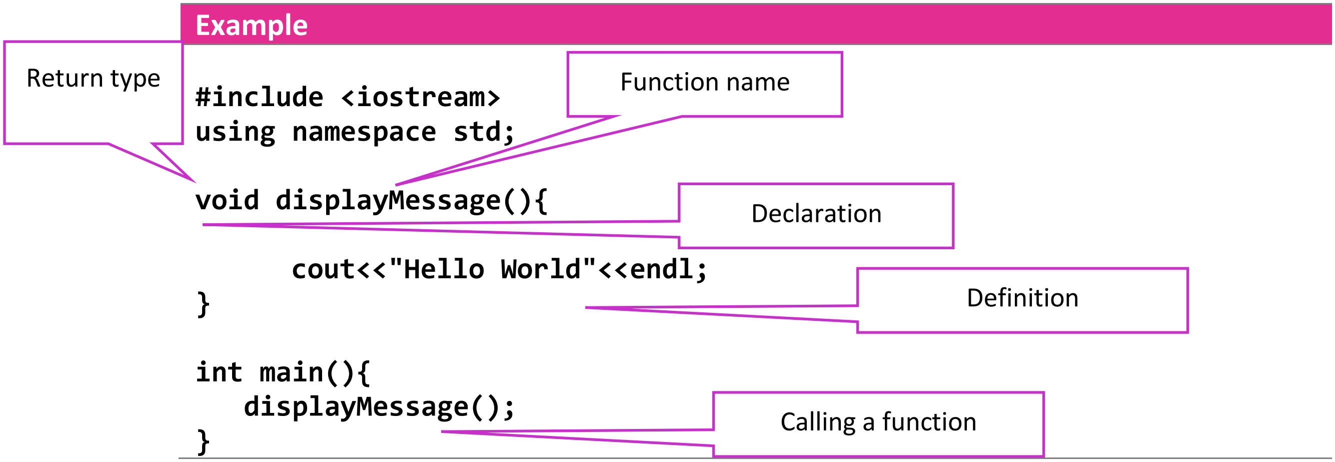
➦ Make changes in the following program so that it uses a switch-case statement in place of the if-elseif- else statement:

```
int legs=0;
 cout << "Enter number of legs ";
 cin >> legs;
 else if(legs == 2)
    cout << "Bird";
 else if(legs == 4)
   cout << "Cat";
 else if(legs == 6)
   cout << "Spider";
 else cout << "I don't know what you are";
```

Function

A function is a collection of statements that performs a specific task.





Example

```
#include <iostream>
using namespace std;

void displayMessage(){

      cout<<"Hello World"<<endl;
}

int main(){
   displayMessage();
}
```

CHECK YOUR KNOWLEDGE

1- Write a program that gives the maximum of three positive integers.

Sample Run:

Enter number 1:15

Enter number 2:70

Enter number 3:23

The maximum of these numbers is: 70

2- Using the loop of for loop, write a program that computes the sum of all odd integers between 0 and 100.

3- Problem - A - Codeforces

4- https://codeforces.com/contest/734/problem/A