Introduction to C++

Basics with Moatasem Elsayed

Content

-introduction
- c++ standards
-hello world
-out
-datatypes
-in
-global vs local
-memory sections
-operators
-if/switch
-for/while/do

Introduction

C++ is a general-purpose programming language that was developed as an extension of the C programming language. It was created by Bjarne Stroustrup in the early 1980s and is widely used for developing a wide range of applications, including system software, desktop applications, games, embedded systems, and more. C++ is known for its flexibility, performance, and ability to support both procedural and object-oriented programming paradigms.

Object-Oriented Programming (OOP): C++ supports classes and objects, enabling developers to build reusable and modular code through encapsulation, inheritance, and polymorphism

Strongly Typed: C++ is a statically-typed language, which means that variable types must be declared before use. This helps catch errors at compile-time and improves performance

Cont ...

Standard Template Library (STL): C++ comes with a powerful STL that provides a collection of data structures (e.g., vectors, lists, maps) and algorithms (e.g., sorting, searching) that can be used in a generic and efficient manner

```cpp
// vector::push_back
#include <iostream>
#include <vector>
int main ()
{
  std::vector<int> myvector;
  int myint;
  std::cout << "Please enter some integers (enter 0 to end):\n";
  do {
    std::cin >> myint;
    myvector.push_back (myint);
  } while (myint);
  std::cout << "myvector stores " << int(myvector.size()) << " numbers.\n";
  return 0;
}
```

Cont ...

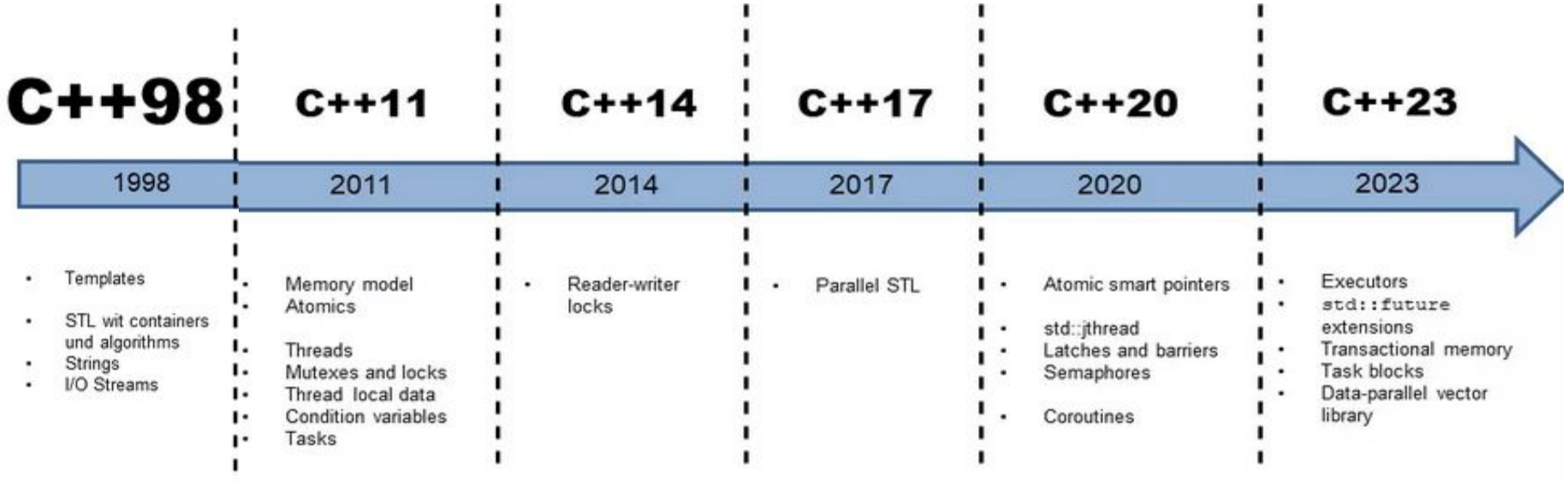
Pointers and Memory Management: C++ provides direct memory manipulation through pointers, allowing developers to manage memory efficiently. However, this also introduces the risk of memory-related bugs, such as segmentation faults.

Multi-Paradigm Language: C++ supports procedural, object-oriented, and generic programming styles, providing developers with flexibility in choosing the best approach for their projects

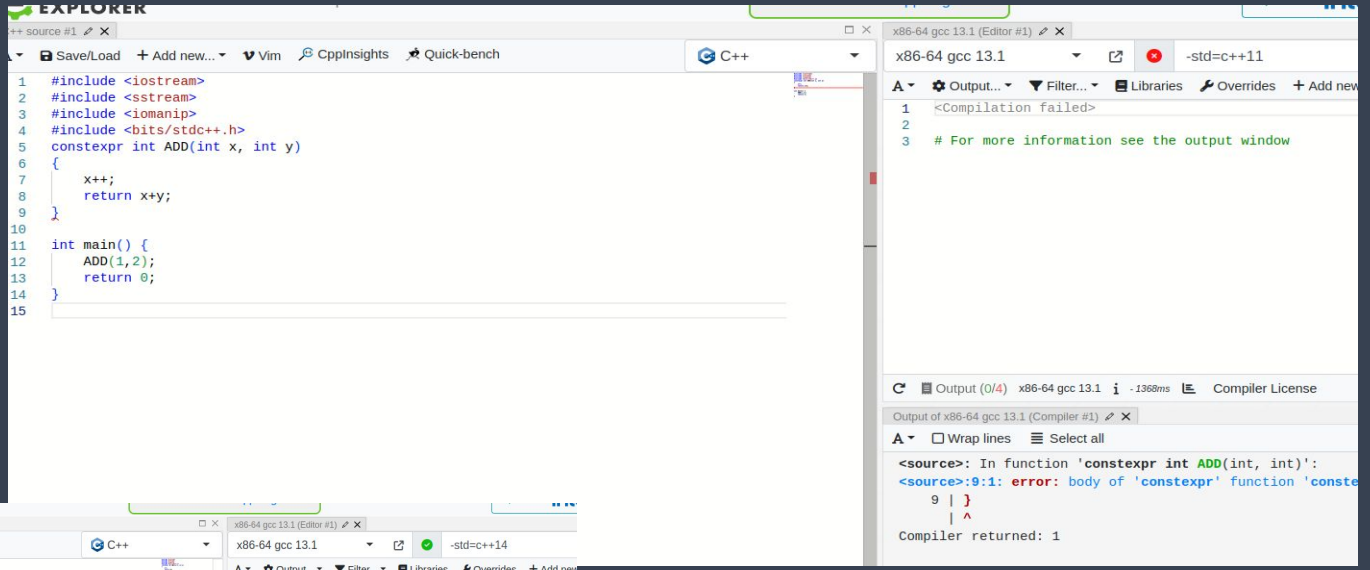
What is C++?

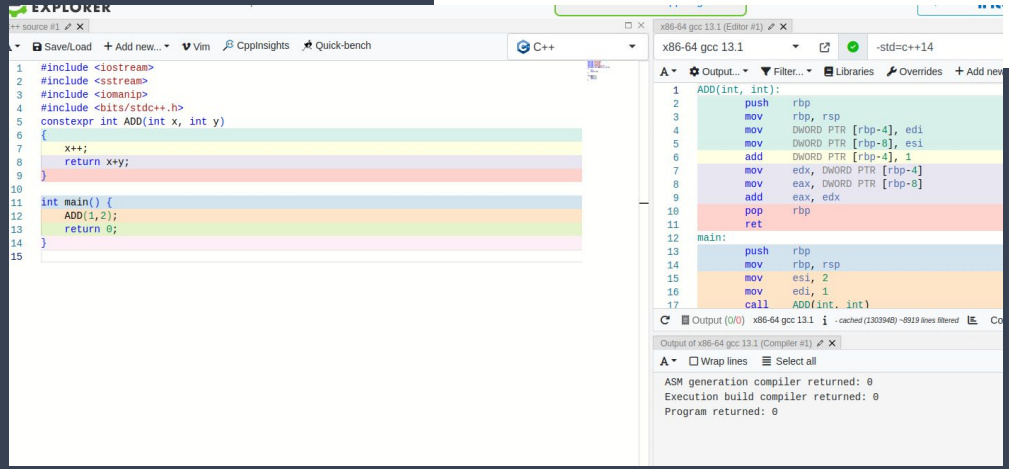
C++ is a general-purpose, object-oriented programming language. It was created by Bjarne Stroustrup at Bell Labs circa 1980. C++ is very similar to C (invented by Dennis Ritchie in the early 1970s). C++ is so compatible with C that it will probably compile over 99% of C programs without changing a line of source code. Though C++ is a lot of well-structured and safer language than C as it OOPs based.

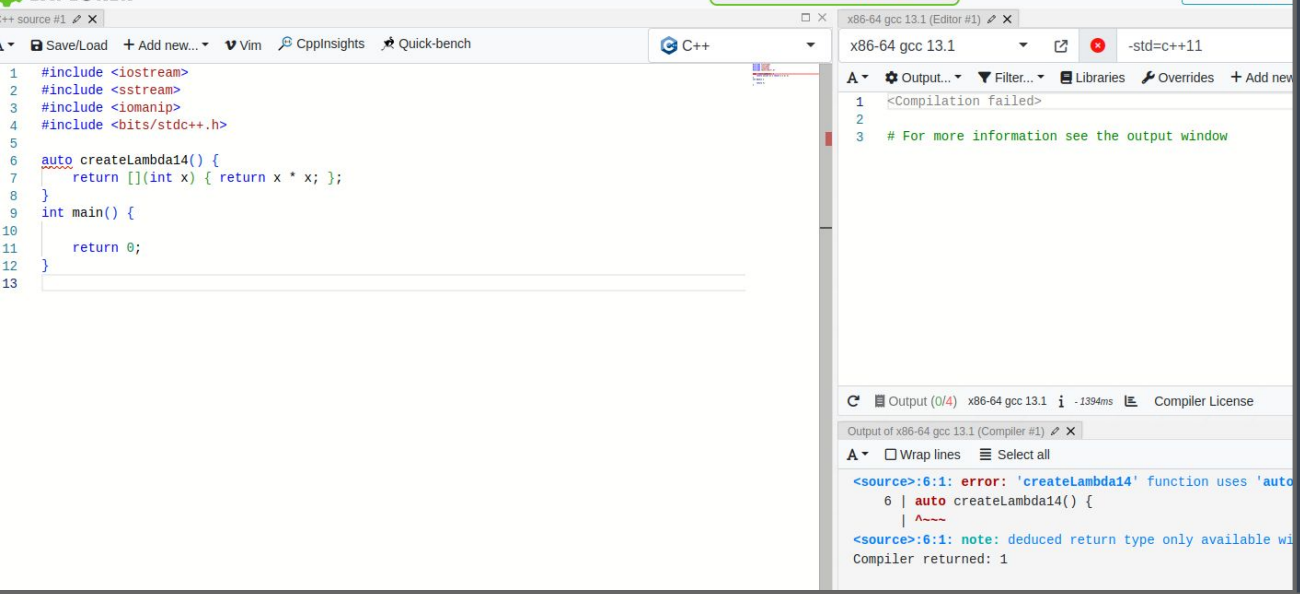
C++ standards

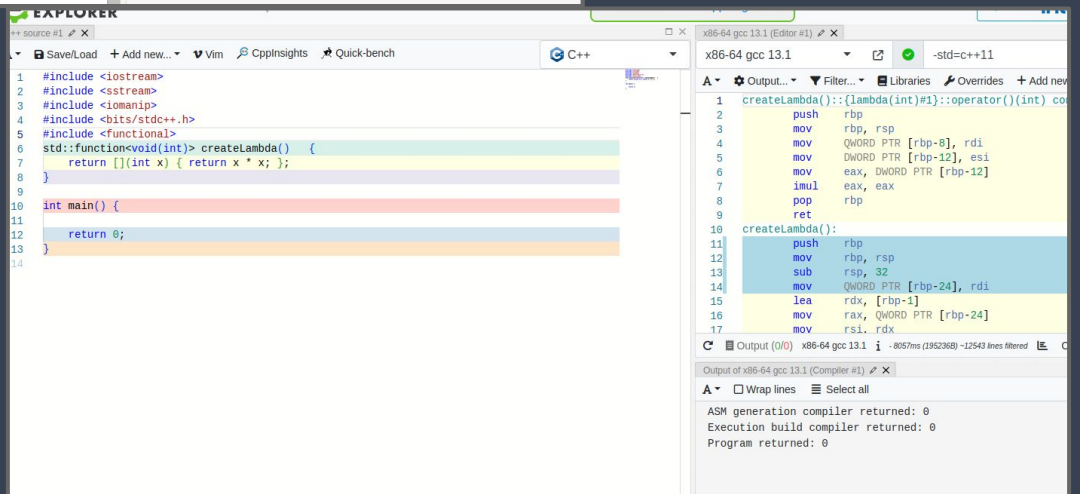

C++98
1998
Templates
STL wit containers und algorithms
Strings
I/O Streams
C++11
2011
Memory model
Atomics
Threads
Mutexes and locks
Thread local data
Condition variables
Tasks
C++14
2014
Reader-writer locks
C++17
2017
Parallel STL
C++20
2020
Atomic smart pointers
std::jthread
Latches and barriers
Semaphores
Coroutines
C++23
2023
Executors
std::future extensions
Transactional memory
Task blocks
Data-parallel vector library

C++ 11 vs c++14


EXPLORER
Save/Load
Add new...
Vim
CppInsights
Quick-bench
C++
x86-64 gcc 13.1
-std=c++11
#include <iostream>
#include <sstream>
#include <iomanip>
#include <bits/stdc++.h>
constexpr int ADD(int x, int y)
{
x++;
return x+y;
}
int main() {
ADD(1,2);
return 0;
}
<Compilation failed>
For more information see the output window
Output (0/4)
Compiler License
Wrap lines
Select all
<source>: In function 'constexpr int ADD(int, int)':
<source>:9:1: error: body of 'constexpr' function 'conste
9 | }
| ^
Compiler returned: 1

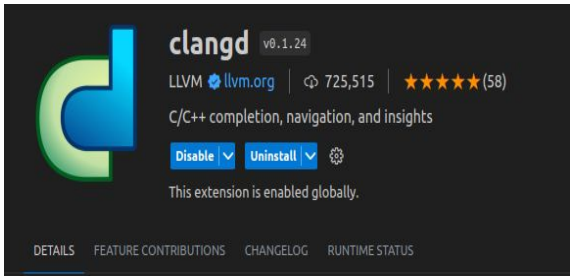

EXPLORER
Save/Load
Add new...
Vim
CppInsights
Quick-bench
C++
x86-64 gcc 13.1
-std=c++14
#include <iostream>
#include <sstream>
#include <iomanip>
#include <bits/stdc++.h>
constexpr int ADD(int x, int y)
{
x++;
return x+y;
}
int main() {
ADD(1,2);
return 0;
}
ADD(int, int):
push rbp
mov rbp, rsp
mov DWORD PTR [rbp-4], edi
mov DWORD PTR [rbp-8], esi
add DWORD PTR [rbp-4], 1
mov edx, DWORD PTR [rbp-4]
mov eax, DWORD PTR [rbp-8]
add eax, edx
pop rbp
ret
main:
push rbp
mov rbp, rsp
mov esi, 2
mov edi, 1
call ADD(int, int)
Output (0/0)
Wrap lines
Select all
ASM generation compiler returned: 0
Execution build compiler returned: 0
Program returned: 0

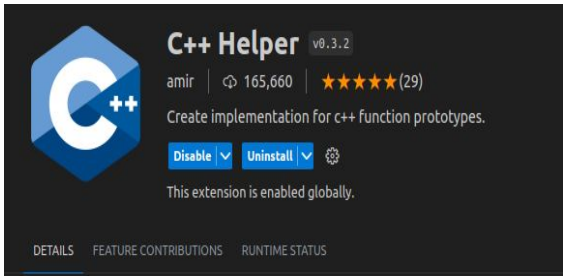

Save/Load
Add new...
Vim
CppInsights
Quick-bench
C++
x86-64 gcc 13.1
-std=c++11
Output...
Filter...
Libraries
Overrides
Add new
#include <iostream>
#include <sstream>
#include <iomanip>
#include <bits/stdc++.h>
auto createLambda14() {
return [](int x) { return x * x; };
}
int main() {
return 0;
}
<Compilation failed>
For more information see the output window
Output (0/4)
x86-64 gcc 13.1
Compiler License
Wrap lines
Select all
<source>:6:1: error: 'createLambda14' function uses 'auto
6 | auto createLambda14() {
| ^~~~
<source>:6:1: note: deduced return type only available wi
Compiler returned: 1

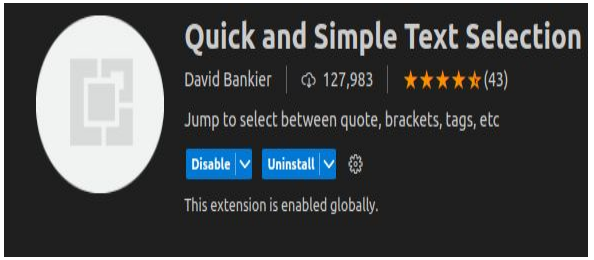

EXPLORER
Save/Load
Add new...
Vim
CppInsights
Quick-bench
C++
x86-64 gcc 13.1
-std=c++11
#include <iostream>
#include <sstream>
#include <iomanip>
#include <bits/stdc++.h>
#include <functional>
std::function<void(int)> createLambda() {
return [](int x) { return x * x; };
}
int main() {
return 0;
}
createLambda()::{lambda(int)#1}::operator()(int) co
push rbp
mov rbp, rsp
mov QWORD PTR [rbp-8], rdi
mov DWORD PTR [rbp-12], esi
mov eax, DWORD PTR [rbp-12]
imul eax, eax
pop rbp
ret
createLambda():
push rbp
mov rbp, rsp
sub rsp, 32
mov QWORD PTR [rbp-24], rdi
lea rdx, [rbp-1]
mov rax, QWORD PTR [rbp-24]
mov rsi, rdx
Output (0/0)
x86-64 gcc 13.1
Wrap lines
Select all
ASM generation compiler returned: 0
Execution build compiler returned: 0
Program returned: 0

Vscode

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ sudo apt-get install build-essential
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.9ubuntu3).
0 upgraded, 0 newly installed, 0 to remove and 57 not upgraded.
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ sudo apt-get install g++
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
g++ is already the newest version (4:11.2.0-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 57 not upgraded.
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$
```



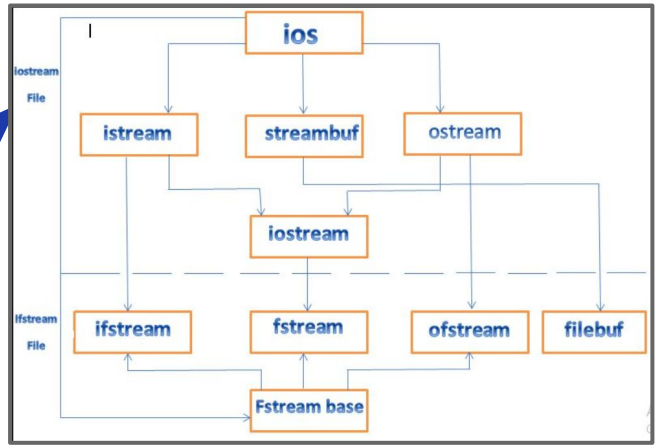





```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ sudo apt-get install cppman
[sudo] password for moatsem:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
cppman is already the newest version (0.5.3+dfsg1-1).
0 upgraded, 0 newly installed, 0 to remove and 57 not upgraded.
```

Hello World

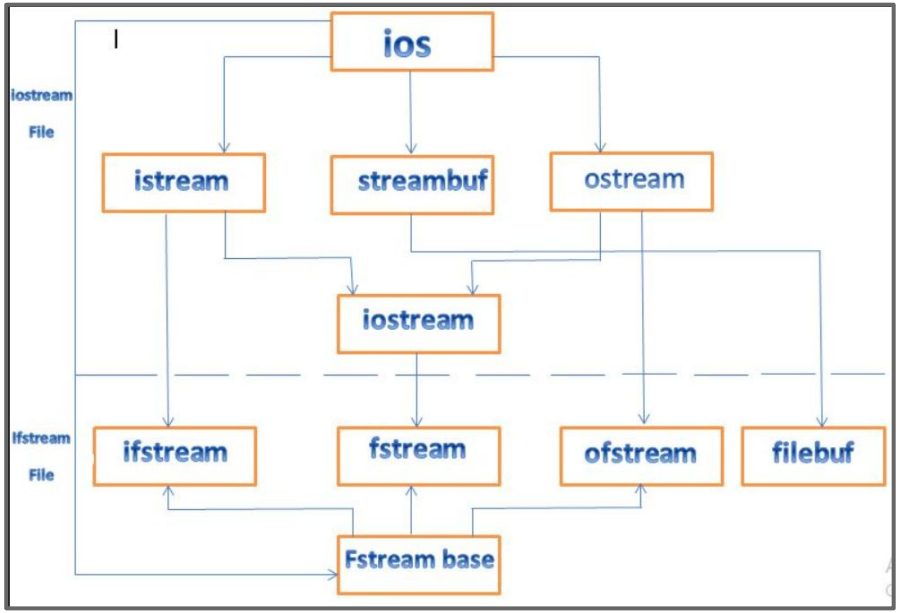
```
namepsace std{
ostream& cout;
};
```



namespace

```cpp
#include <iostream>

int main() {
    std::cout << "Hello, World!" << std::endl;
    return 0;
}
```

By End of C++ Course Check this slide




```cpp
#include <iostream>

int main() {
    std::ostream& output = std::cout; // Get a reference to the standard output stream (cout)

    output << "Hello, World!" << std::endl; // Print "Hello, World!" to the output stream

    return 0;
}
```

Comments

```
// Single line comment

/*
 * Multi-line comment syntax
 * Comments help us to understand program later easily
 * Will you write comments while writing programs?
 */
```

1- multiple comments

/*

/*

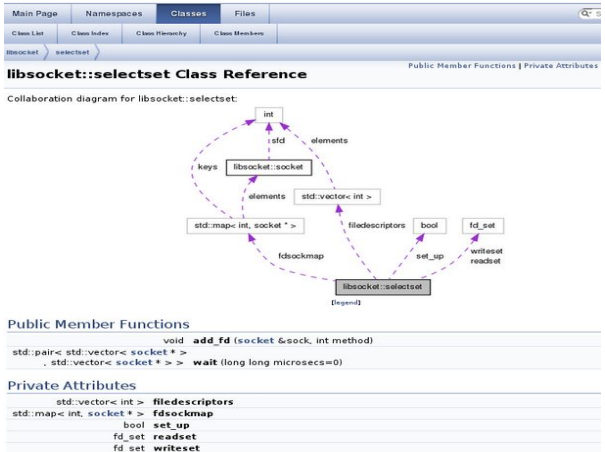
*/

*/

two comments inside each other is not allowed

2-tools using comments style like doxygen

```
//!  ------------------------------------------------------------
//! @brief startKeepLive is mainloop to lunch main thread
//! @param void
//! @return void
//! @exception N/A
//! @todo N/A
```



Display (out)

Task : std::flush

```cpp
#include <iostream>

int main() {
  std::cout << "Hello world" << std::endl;
  std::cout << "Hello world\n";
  printf(format: "Hello world\n");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ g++ test.cpp -Wall
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ ./a.out
Hello world
Hello world
Hello world
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$
```

- %d integer number
- %f floating number
- %c characters
- %o The unsigned octal format specifier.
- %s The string format specifier.
- %u The unsigned integer format specifier.
- %x The unsigned hexadecimal format specifier a b c
- %X The unsigned hexadecimal format specifier A B C

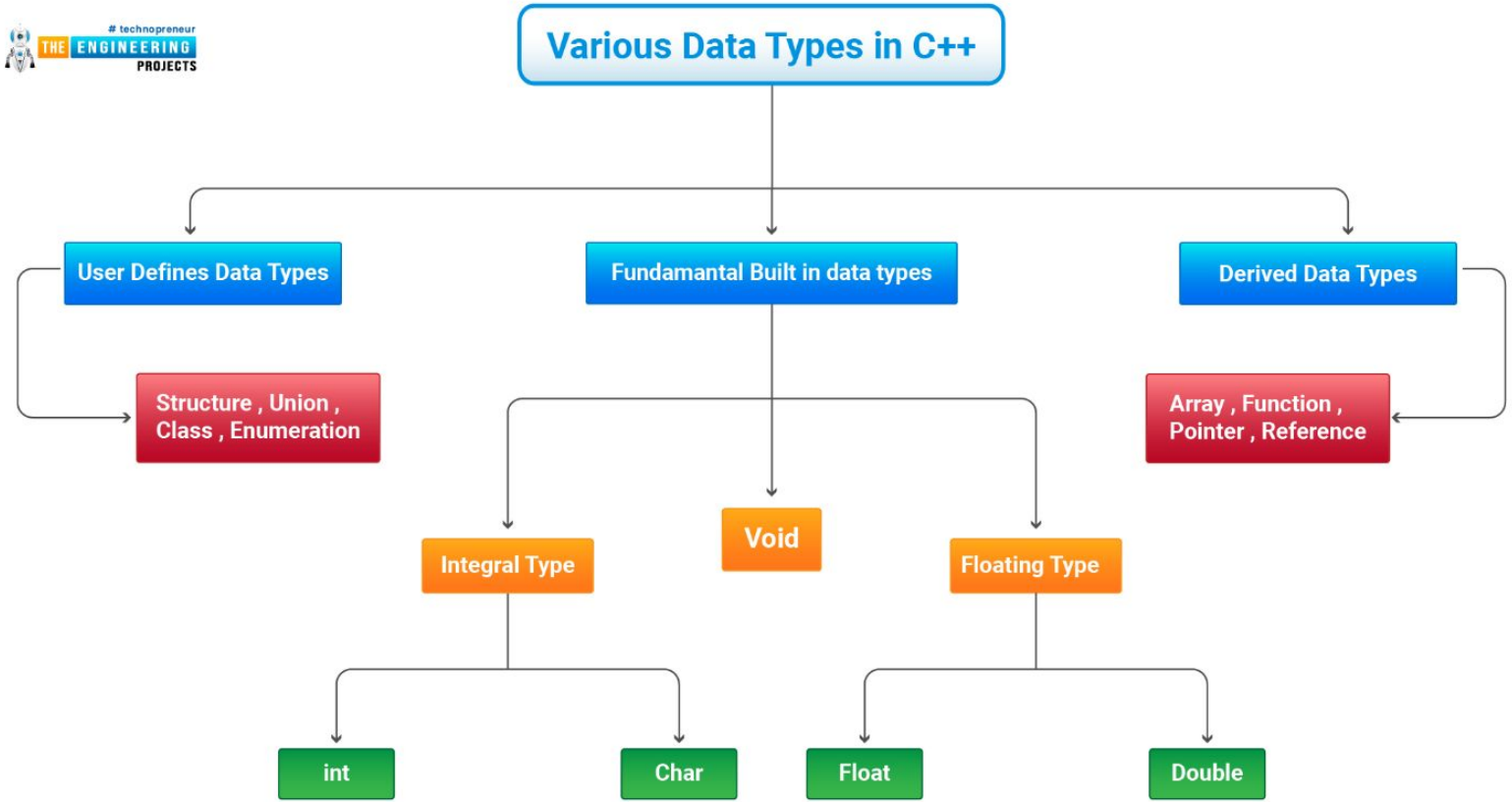
Check slide 15

```cpp
int main() {
  int x = 5;
  std::cout << "Value : " << x << std::endl;
  std::cout << "Value " << x << "\n";
  printf(format: "Value %d\n", x);
  return 0;
}
```

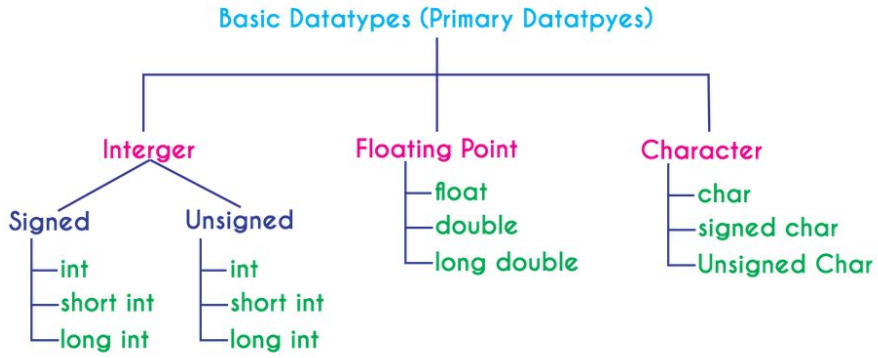
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_
Value : 5
Value 5
Value 5
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_
```

Data Types


Various Data Types in C++
User Defines Data Types
Fundamantal Built in data types
Derived Data Types
Structure , Union , Class , Enumeration
Array , Function , Pointer , Reference
Integral Type
Void
Floating Type
int
Char
Float
Double

Recap on C Style on Data types




```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ g++ datatypes.cpp
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ ./a.out
Size of char: 1 bytes
Size of short: 2 bytes
Size of int: 4 bytes
Size of long: 8 bytes
Size of long long: 8 bytes
Size of float: 4 bytes
Size of double: 8 bytes
Size of long double: 16 bytes
Size of bool: 1 bytes
Size of std::string: 32 bytes
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$
```

```cpp
#include <cstdio>

int main() {
  int x = 10;
  printf("int: %d\n", x);
  char c = 'A';
  printf("char: %c\n", c);
  short s = 123;
  printf("short: %hd\n", s);
  long l = 1234567890;
  printf("long: %ld\n", l);
  long long ll = 1234567890123456789;
  printf("long long: %lld\n", ll);
  float f = 3.14f;
  printf("float: %f\n", f);
  double d = 3.141592653589793;
  printf("double: %lf\n", d);
  long double ld = 3.141592653589793;
  printf("long double: %Lf\n", ld);
  bool b = true;
  printf("bool: %d\n", b);

  return 0;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ g++ test.cpp -Wall
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ ./a.out
10
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ g++ test.cpp -Wall
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ ./a.out
int: 10
char: A
short: 123
long: 1234567890
long long: 1234567890123456789
float: 3.140000
double: 3.141593
long double: 3.141593
bool: 1
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$
```

arm

`char`	8	1 (byte-aligned)	0 to 255 (unsigned) by default. -128 to 127 (signed) when compiled with `--signed_chars`.
`signed char`	8	1 (byte-aligned)	-128 to 127
`unsigned char`	8	1 (byte-aligned)	0 to 255
`(signed) short`	16	2 (halfword-aligned)	-32,768 to 32,767
`unsigned short`	16	2 (halfword-aligned)	0 to 65,535
`(signed) int`	32	4 (word-aligned)	-2,147,483,648 to 2,147,483,647
`unsigned int`	32	4 (word-aligned)	0 to 4,294,967,295
`(signed) long`	32	4 (word-aligned)	-2,147,483,648 to 2,147,483,647
`unsigned long`	32	4 (word-aligned)	0 to 4,294,967,295
`(signed) long long`	64	8 (doubleword-aligned)	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
`unsigned long long`	64	8 (doubleword-aligned)	0 to 18,446,744,073,709,551,615

Table 1. Data Types on AVR 8-bit Microcontrollers in <stdint.h>

Data type		Size
signed char / unsigned char	int8_t / uint8_t	8-bit
signed int / unsigned int	int16_t / uint16_t	16-bit
signed long / unsigned long	int32_t / uint32_t	32-bit
signed long long / unsigned long long	int64_t / uint64_t	64-bit

The data type size and range depend a lot on the compiler. However, the code that the compiler compiles is targeted for some specific types of Microcontrollers or Microprocessors.

One single compiler can provide support for multiple targets or processors. The compiler then defines the size for the available data types on the basis of the selected target.

In simpler words, the size of any data type is directly dependent on the compiler along with the target processor

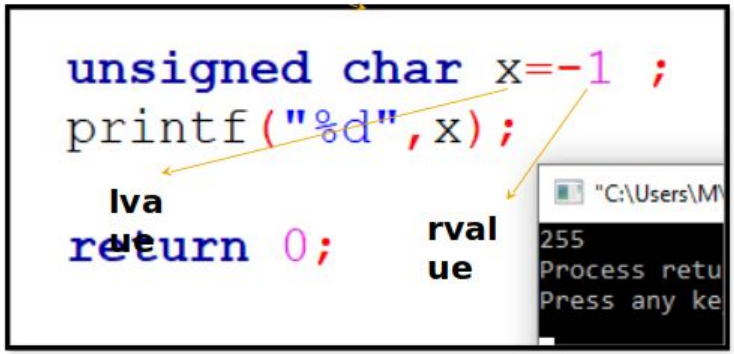
Data Type	Size (bytes)	Size (bits)	Value Range
unsigned char	1	8	0 to 255
signed char	1	8	-128 to 127
char	1	8	either
unsigned short	2	16	0 to 65,535
short	2	16	-32,768 to 32,767
unsigned int	4	32	0 to 4,294,967,295
int	4	32	-2,147,483,648 to 2,147,483,647
unsigned long	8	64	0 to 18,446,744,073,709,551,616
long	8	64	–9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
unsigned long long	8	64	0 to 18,446,744,073,709,551,616
long long	8	64	–9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4	32	3.4E +/- 38 (7 digits)
double	8	64	1.7E +/- 308 (15 digits)
long double	8	64	1.7E +/- 308 (15 digits)

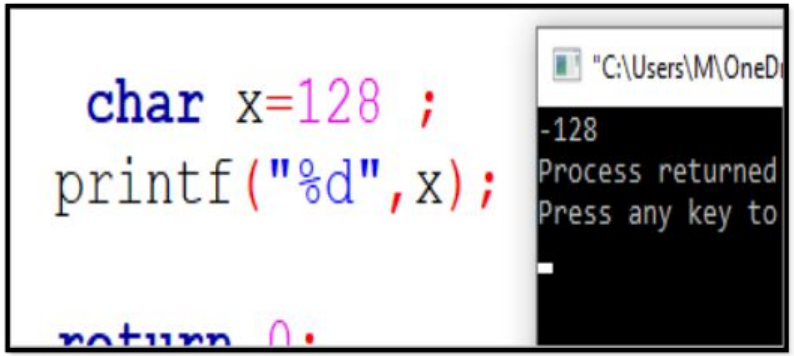
We Have Issue here !

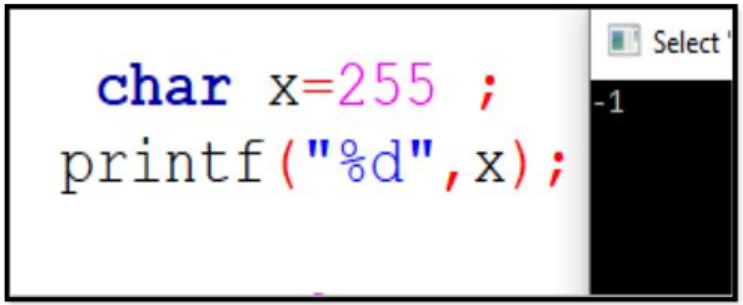
```
5   //   printf("int: %d\n", x);
6   char c = 'A';
7   printf(format: "char A: %c\n", c);
8   char s = 123;
9   printf(format: "char s: %d\n", s);
10
11  return 0;
12  }
```

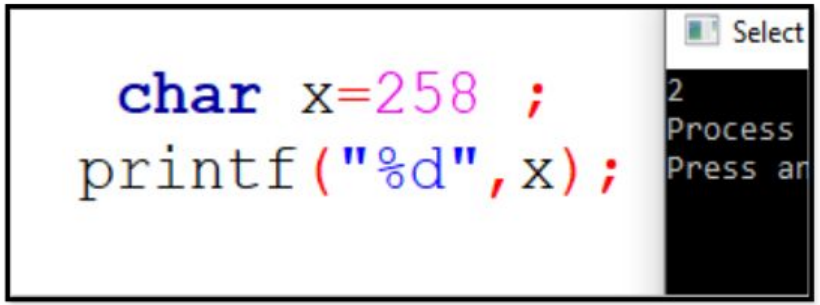
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ g++ test.cpp -Wall
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ ./a.out
char A: A
char s: 123
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$
```

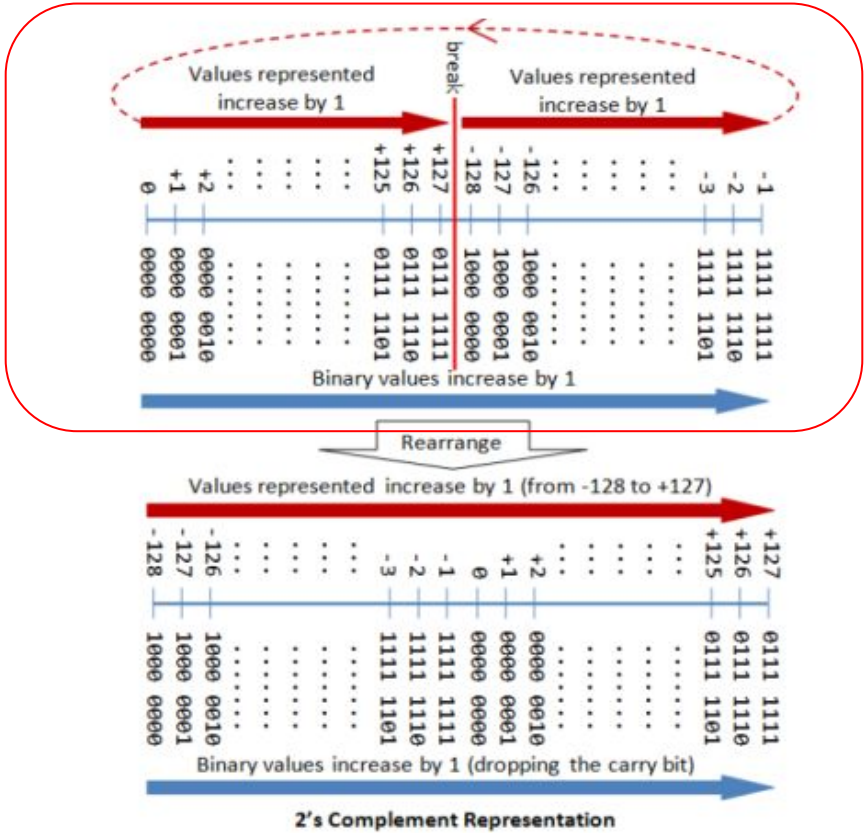




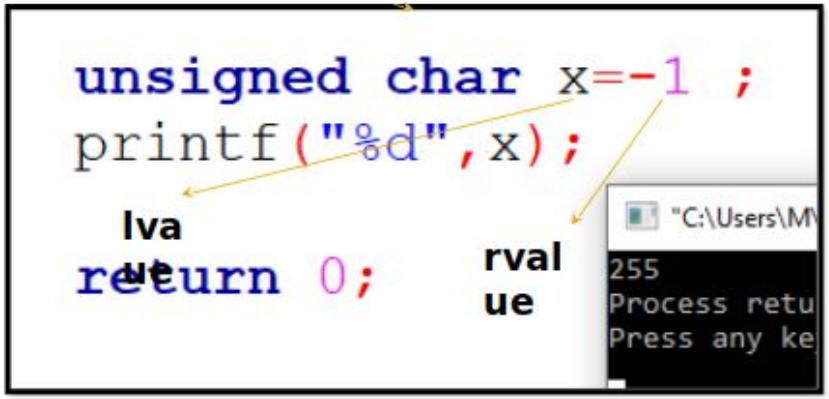


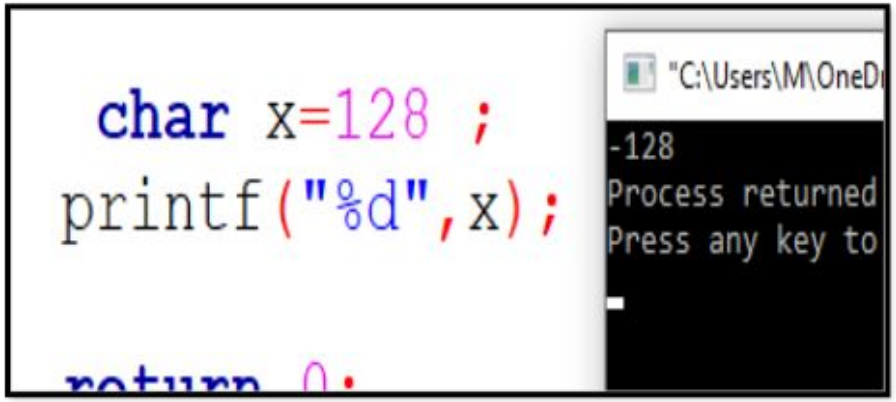


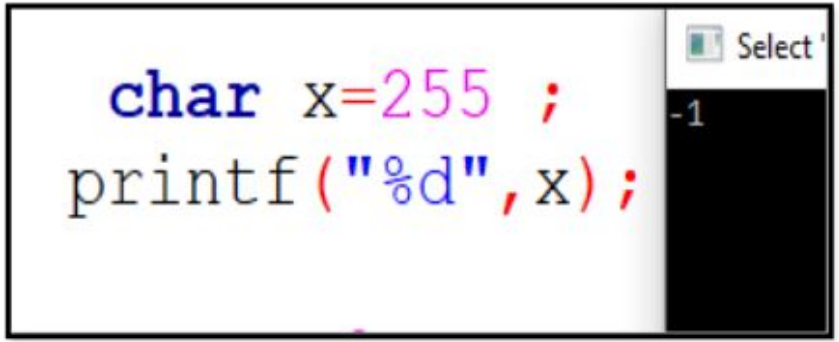
Signed data type

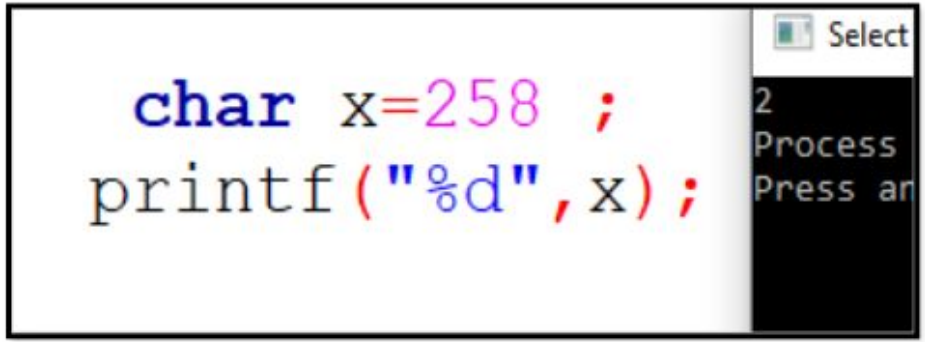


Examples


unsigned char x=-1 ;
printf("%d",x);
lva
ue
return 0;
rval
ue
"C:\Users\M\
255
Process retu
Press any ke


char x=128 ;
printf("%d",x);
"C:\Users\M\OneD
-128
Process returned
Press any key to


char x=255 ;
printf("%d",x);
Select
-1


char x=258 ;
printf("%d",x);
Select
2
Process
Press an

Input

```cpp
int main() {
  int x;
  std::cout << "Enter an int: ";
  std::cin >> x;
  std::cout << "int: value is " << x << std::endl;
  char c;
  std::cout << "Enter a char: ";
  std::cin >> c;
  std::cout << "char: value is " << c << std::endl;
  short s;
  std::cout << "Enter a short: ";
  std::cin >> s;
  std::cout << "short: value is " << s << std::endl;
  long l;
  std::cout << "Enter a long: ";
  std::cin >> l;
  std::cout << "long: value is " << l << std::endl;
  long long ll;
  std::cout << "Enter a long long: ";
  std::cin >> ll;
  std::cout << "long long: value is " << ll << std::endl;
  float f;
  std::cout << "Enter a float: ";
  std::cin >> f;
  std::cout << "float: value is " << f << std::endl;
  double d;
  std::cout << "Enter a double: ";
  std::cin >> d;
  std::cout << "double: value is " << d << std::endl;
  long double ld;
  std::cout << "Enter a long double: ";
  std::cin >> ld;
  std::cout << "long double: value is " << ld << std::endl;
  bool b;
  std::cout << "Enter a bool (0 or 1): ";
  std::cin >> b;
  std::cout << "bool: value is " << b << std::endl;

  return 0;
```

```cpp
float y;
int x;
char z;
printf("please enter value");
scanf("%d%f",&x,&y);
printf("the value of x=%d and y=%f",x,y);
scanf(" %c",&z);
printf("take care with space before %c ",z);
```

```
"E:\self learnin\embded system\my presentaion\c programming\assisgments +codes\valeo interview code\nonrepeatednumbers\nam
please enter value
5
6.5
the value of x=5 and y=6.500000
a
take care with space before a
```

manipulators

For more check `<iomanip>`

```cpp
int main() {
    int num2 = 42;
    std::cout << "Number: " << std::setw(6) << num2 << std::endl;

    double pi = 3.14159265358979323846;
    std::cout << "Pi: " << std::setprecision(4) << pi << std::endl;
    double num3 = 123.456789;
    std::cout << "Number: " << std::scientific << num3 << std::endl;

    int num = 42;
    std::cout << "Hex: " << std::hex << num << std::endl;
    std::cout << "Oct: " << std::oct << num << std::endl;
    std::cout << "Dec: " << std::dec << num << std::endl;
    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ g++ manipualtor.cpp
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ ./a.out
Number:     42
Pi: 3.142
Number: 1.2346e+02
Hex: 2a
Oct: 52
Dec: 42
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$
```

```cpp
bool b;
std::cout << "Enter a bool (0 or 1): ";
std::cin >> b;
std::cout << "bool: value is " << std::boolalpha << b << std::endl;

return 0;
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ `^C
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ g++ test.cpp -Wall
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ ./a.out
Enter a bool (0 or 1): 1
bool: value is true
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ ./a.out
Enter a bool (0 or 1): 0
bool: value is false
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$
```

Global Vs Local

```c
int y; /*** y is global variable ***/

int main()
{
    int x; /**** x is local variable *****/
    printf("hello world ");
    return 0;
}
```

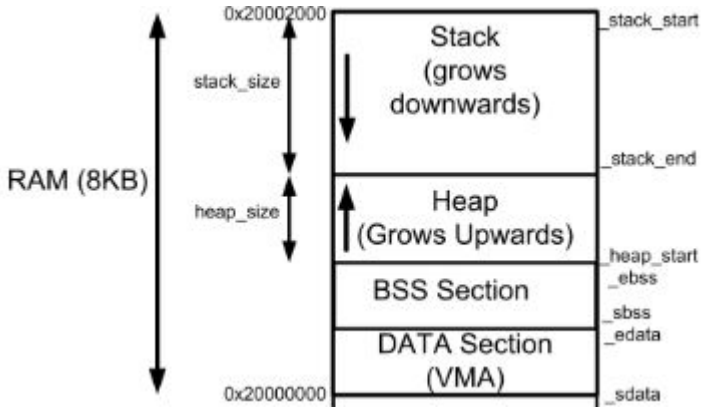
```cpp
int main() {
  int x;
  std::cout << x << std::endl;
  return 0;
}
```

Rule MS_1.0: don't let compiler to initialize the local variables

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ g++ test.cpp -Wall
test.cpp: In function ‘int main()’:
test.cpp:6:16: warning: ‘x’ is used uninitialized [-Wuninitialized]
    6 |   std::cout << x << std::endl;
      |                ^
test.cpp:5:7: note: ‘x’ was declared here
    5 |   int x;
      |       ^
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ ./a.out
32730
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$
```

Memory sections

```
7         std::cout << "Hello World" << This_x_Stack << std::endl;
(gdb) disassemble /s main
Dump of assembler code for function main():
memory_sec.cpp:
5       int main() {
   0x00005555555551c9 <+0>:     endbr64
   0x00005555555551cd <+4>:     push   %rbp
   0x00005555555551ce <+5>:     mov    %rsp,%rbp
   0x00005555555551d1 <+8>:     sub    $0x10,%rsp

6         int This_x_Stack = 0;
   0x00005555555551d5 <+12>:    movl   $0x0,-0x4(%rbp)

7         std::cout << "Hello World" << This_x_Stack << std::endl;
=> 0x00005555555551dc <+19>:    lea    0xe21(%rip),%rax        # 0x555555556004
   0x00005555555551e3 <+26>:    mov    %rax,%rsi
   0x00005555555551e6 <+29>:    lea    0x2e53(%rip),%rax        # 0x555555558040
   0x00005555555551ed <+36>:    mov    %rax,%rdi
   0x00005555555551f0 <+39>:    call   0x5555555550a0 <_ZStlsISt11char_traitsIcEE
   0x00005555555551f5 <+44>:    mov    %rax,%rdx
   0x00005555555551f8 <+47>:    mov    -0x4(%rbp),%eax
   0x00005555555551fb <+50>:    mov    %eax,%esi
   0x00005555555551fd <+52>:    mov    %rdx,%rdi
   0x0000555555555200 <+55>:    call   0x5555555550d0 <_ZNSolsEi@plt>
```




```
#include <string>
int This_x_Data_Section;
int This_x_BSS = 10;
int main() {
  int This_x_Stack = 0;
  std::cout << "Hello World" << This_x_Stack << std::endl;

  return 0;
}
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ objdump -t ./a.out --demangle | grep " This"
0000000000004154 g     O .bss   0000000000000004              This_x_Data_Section
0000000000004010 g     O .data  0000000000000004              This_x_BSS
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$
```

Base Pointer (`%rbp`): The `%rbp` register is used as a reference point within a function stack frame. It points to the base of the current function's stack

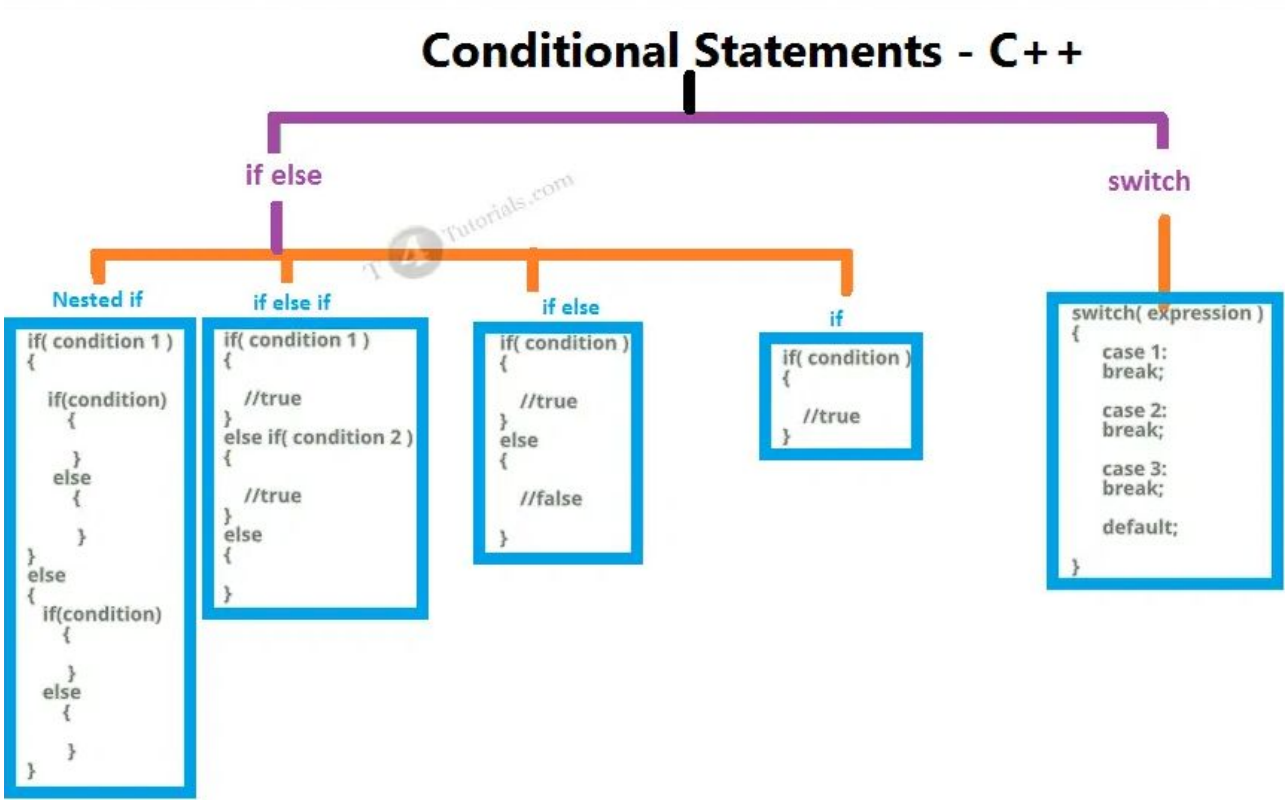
Operators

operators.cpp > ...

```cpp
    std::cout << "Arithmetic Operators:" << std::endl;
    std::cout << "a + b = " << a + b << std::endl;
    std::cout << "a - b = " << a - b << std::endl;
    std::cout << "a * b = " << a * b << std::endl;
    std::cout << "a / b = " << a / b << std::endl;
    std::cout << "a % b = " << a % b << std::endl;

    // Comparison Operators
    std::cout << "\nComparison Operators:" << std::endl;
    std::cout << "a == b: " << (a == b) << std::endl;
    std::cout << "a != b: " << (a != b) << std::endl;
    std::cout << "a > b: " << (a > b) << std::endl;
    std::cout << "a < b: " << (a < b) << std::endl;
    std::cout << "a >= b: " << (a >= b) << std::endl;
    std::cout << "a <= b: " << (a <= b) << std::endl;

    // Logical Operators
    bool x = true, y = false;
    std::cout << "\nLogical Operators:" << std::endl;
    std::cout << "x && y: " << (x && y) << std::endl;
    std::cout << "x || y: " << (x || y) << std::endl;
    std::cout << "!x: " << !x << ", !y: " << !y << std::endl;

    // Bitwise Operators
    unsigned int num1 = 5, num2 = 3;
    std::cout << "\nBitwise Operators:" << std::endl;
    std::cout << "num1 & num2: " << (num1 & num2) << std::endl;
    std::cout << "num1 | num2: " << (num1 | num2) << std::endl;
    std::cout << "num1 ^ num2: " << (num1 ^ num2) << std::endl;
    std::cout << "~num1: " << ~num1 << std::endl;
    std::cout << "num1 << 1: " << (num1 << 1) << std::endl;
    std::cout << "num1 >> 1: " << (num1 >> 1) << std::endl;

    // Assignment Operators
    int c = 7;
    std::cout << "\nAssignment Operators:" << std::endl;
    c += 3;
    std::cout << "c += 3: " << c << std::endl;
    c -= 2;
    std::cout << "c -= 2: " << c << std::endl;
    c *= 4;
```

OPERATOR	TYPE	ASSOCIAVITY
() [] . ->		left-to-right
++ -- + - ! ~ (*type*) * & sizeof	Unary Operator	right-to-left
* / %	Arithmetic Operator	left-to-right
+ -	Arithmetic Operator	left-to-right
<< >>	Shift Operator	left-to-right
< <= > >=	Relational Operator	left-to-right
== !=	Relational Operator	left-to-right
&	Bitwise AND Operator	left-to-right
^	Bitwise EX-OR Operator	left-to-right
|	Bitwise OR Operator	left-to-right
&&	Logical AND Operator	left-to-right
||	Logical OR Operator	left-to-right
? :	Ternary Conditional Operator	right-to-left
= += -= *= /= %= &= ^= |= <<= >>=	Assignment Operator	right-to-left
,	Comma	left-to-right

Conditional



Examples

```cpp
#include <iostream>

int main() {
    int age;
    std::cout << "Enter your age: ";
    std::cin >> age;

    if (age >= 18) {
        std::cout << "You are eligible to vote." << std::endl;
    }

    return 0;
}
```

```cpp
#include <iostream>

int main() {
    int score;
    std::cout << "Enter your exam score: ";
    std::cin >> score;

    if (score >= 60) {
        std::cout << "You passed the exam." << std::endl;
    } else {
        std::cout << "You did not pass the exam." << std::endl;
    }

    return 0;
}
```

```cpp
#include <iostream>

int main() {
    int num;
    std::cout << "Enter a number: ";
    std::cin >> num;

    if (num > 0) {
        std::cout << "The number is positive." << std::endl;
    } else if (num < 0) {
        std::cout << "The number is negative." << std::endl;
    } else {
        std::cout << "The number is zero." << std::endl;
    }

    return 0;
}
```

```cpp
#include <iostream>

int main() {
    int num;
    std::cout << "Enter a number: ";
    std::cin >> num;

    if (num >= 0) {
        if (num == 0) {
            std::cout << "The number is zero." << std::endl;
        } else {
            std::cout << "The number is positive." << std::endl;
        }
    } else {
        std::cout << "The number is negative." << std::endl;
    }

    return 0;
}
```

```cpp
#include <iostream>

int main() {
    int num;
    std::cout << "Enter a number: ";
    std::cin >> num;

    if (num > 0 && num <= 100) {
        std::cout << "The number is in the range 1 to 100." << std::en
    } else if (num <= 0 || num > 100) {
        std::cout << "The number is outside the range 1 to 100." << st
    }

    return 0;
}
```

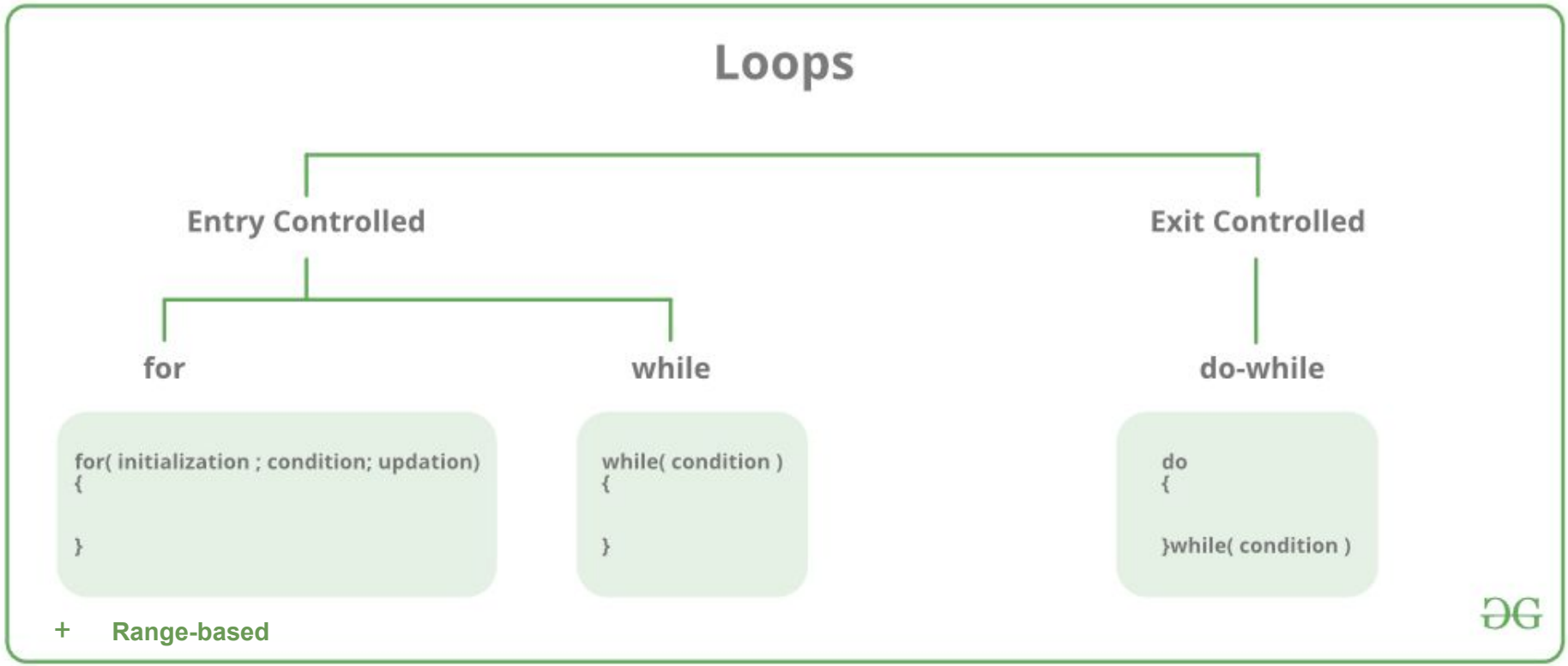
Switch

1- switch(integral)
2- case constant :
3- break;
4- case 1 case 2 case 3:
5- declare variable after case :

```cpp
#include <iostream>

int main() {
    int choice;

    std::cout << "Menu:" << std::endl;
    std::cout << "1. Print Hello" << std::endl;
    std::cout << "2. Print World" << std::endl;
    std::cout << "3. Exit" << std::endl;
    std::cout << "Enter your choice: ";
    std::cin >> choice;

    switch (choice) {
        case 1:
            std::cout << "Hello" << std::endl;
            break;
        case 2:
            std::cout << "World" << std::endl;
            break;
        case 3:
            std::cout << "Exiting..." << std::endl;
            break;
        default:
            std::cout << "Invalid choice" << std::endl;
    }

    return 0;
}
```

```cpp
using namespace std;

int main(){
    int score;

    //Vraag de score
    cout << "Score:";
    cin >> score;
    const int ZERO=0;
    //Switch
    switch(score){
        case ZERO:
            cout << "a";
            break;
        case  9:
```

Loops



\+ Range-based

Examples

```cpp
#include <iostream>

int main() {
    for (int i = 1; i <= 5; ++i) {
        std::cout << "Iteration " << i << std::endl;
    }

    return 0;
}
```

```cpp
#include <iostream>

int main() {
    int count = 0;
    while (count < 5) {
        std::cout << "Count: " << count << std::endl;
        ++count;
    }

    return 0;
}
```

```cpp
#include <iostream>

int main() {
    int num = 5;
    do {
        std::cout << "Num: " << num << std::endl;
        --num;
    } while (num > 0);

    return 0;
}
```

```cpp
#include <iostream>

int main() {
    for (int i = 1; i <= 3; ++i) {
        for (int j = 1; j <= 3; ++j) {
            std::cout << i << " * " << j << " = " << i * j << std::endl;
        }
    }

    return 0;
}
```

Break , continue

```cpp
#include <iostream>

int main() {
    for (int i = 1; i <= 5; ++i) {
        if (i == 3) {
            std::cout << "Skipping iteration " << i << std::endl;
            continue;
        }
        if (i == 5) {
            std::cout << "Breaking loop at iteration " << i << std::endl;
            break;
        }
        std::cout << "Iteration " << i << std::endl;
    }

    return 0;
}
```

Ranged based c++11

```cpp
#include <iostream>
#include <vector>

int main() {
    std::vector<int> numbers = {1, 2, 3, 4, 5};
    for (int num : numbers) {
        std::cout << "Number: " << num << std::endl;
    }

    return 0;
}
```

Tasks

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/02_C++/01_introduction$ ./a.out
ASCII Code Table:
+------+-------+
| Char | ASCII |
+------+-------+
|      |     0 |
|      |     1 |
|      |     2 |
|      |     3 |
|      |     4 |
|      |     5 |
|      |     6 |
|      |     7 |
|      |     8 |
|      |     9 |
|      |    10 |
|      |    11 |
|      |    12 |
|      |    13 |
|      |    14 |
```

```
|   @  |   64  |
|   A  |   65  |
|   B  |   66  |
|   C  |   67  |
|   D  |   68  |
|   E  |   69  |
|   F  |   70  |
|   G  |   71  |
|   H  |   72  |
|   I  |   73  |
|   J  |   74  |
|   K  |   75  |
```

1-Create a table for Ascll code

2- maximum number between three values
3-RIGHT angle triangle
4-decide the letter is vowel or not
5-multiplication table
6-summation the digits of integer entered by user
7-change from decimal to binary and vice versa

Use `bitset`

```
moatsem@moatsem-IdeaPad-Gaming-3-15I
Enter a decimal number: 12
Binary representation: 00001100
Enter a binary number: 00001100
Decimal representation: 12
moatsem@moatsem-IdeaPad-Gaming-3-15I
```

```cpp
// Input from the user
std::cout << "Enter an integer: ";
std::cin >> num;

// Convert the integer to a string
std::string numStr = std::to_string(val: num);

// Calculate the sum of digits using string manipulation
for (char digitChar : numStr) {
  int digit = digitChar - '0'; // Convert character to integer
  sum += digit;
}

// Output the result
std::cout << "Sum of digits of " << num << " is: " << sum << std::endl;
```