

Chapter # 2

C++ Programming Basics



Lecture # 1

Instructor: Engr. Nauman
Ahmad Tariq

Objectives



- C++ Program Structure.
- Variables.
- Input/output with “*cin*” and “*cout*”
- Operators
 - Arithmetic Operators.
 - Assignment & Increment Operators.

```
int x;
int y;
int z;
int main() {
    x = 1;
    y = 2;
    z = x + y;
    return 0;
}
```

Language 1 source code

```
public class Main {
    public static void main(String[] args) {
        int x = 1;
        int y = 2;
        int z = x + y;
        System.out.println(z);
    }
}
```

Language 2 source code

Compiler front-end for language 1

Compiler front-end for language 2



Non-optimized intermediate code

Non-optimized intermediate code

Intermediate code optimizer

Optimized intermediate code

Target-1 Code Generator

Target-2 Code Generator

Target-1 machine code

Target-2 machine code



Simple C++ Program



```
#include<iostream>
using namespace std;
int main()
{
cout<<"This is my 1st C++ program";
return 0;
}
```

Whitespace



```
#include<iostream>
```

```
using
```

```
namespace std;
```

```
int main() { cout
```

```
<<
```

```
“This is my 1st C++ program”;
```

```
return
```

```
0; }
```

Comments



● Single Line Comment

```
//This is single line comment
```

Or

```
//This is single
```

```
//Line comment
```

● Multi Line Comment

```
/*This is
```

```
Multi line
```

```
Comment*/
```

Comments Contd...



```
//demonstrates Comments
#include<iostream>
using namespace std;
//main function
int main()
{
cout<<"This is my 1st C++ program";
return 0;
}/* end of
program*/
```

Integer Variables



Type	Bits	Range	Syntax
int (32 bit system)	32 bits	-2,147,483,648 → 2,147,483,647	int var1;
int (16 bit system)	16 bits	-32,768 → 32767	int var1;
long	32 bits	-2,147,483,648 → 2,147,483,647	long int var1; or long var1;
short	16 bits	-32,768 → 32767	short var1;

Integer Variables



```
//demonstrates integer variables
#include<iostream>
using namespace std;
int main()
{
int var1;    //define var1
int var2=20; //20 is integer constant
var1=var2+10;
cout<<"value of var1="<<var1;
return 0;
}/* end of program*/
```

endl Manipulator



```
//endl Manipulator
#include<iostream>
using namespace std;
int main()
{
int var1; //declare var1
int var2=20; //20 is integer constant
var1=var2+10;
cout<<"value of var1="<<var1<<endl;
cout<<"value of var2="<<var2;
return 0;
}/* end of program*/
```

Common Escape Sequences



Escape Sequence	Character
\a	Beep
\b	Backspace
\f	Form feed
\n	New line
\r	Return
\t	Tab
\\	Backslash
\'	Single quotation mark
\"	Double quotation marks
\xdd	Hexadecimal Notations

Escape Sequences (Contd...)



- `cout<<"This is\b my first C++ program";`
This i my first C++ program
- `cout<<"This is\n my first C++ program";`
This is
my first C++ program
- `cout<<"This is my first \r C++ program";`
C++ programirst
- `cout<<"This is\t my first C++ program";`
This is my first C++ program

Escape Sequences (Contd...)



- `cout<<"This is\\ my first C++ program";`
This is \ my first C++ program
- `cout<<"This is\' my first C++ \' program";`
This is ' my first C++ ' program
- `cout<<"\"This is my first C++ program\"";`
This is my first C++ program
- `cout<<"\x128";`
<

Character Variables & Constants



- **Character Variable**

Range: -128 → 127

Memory: 1 byte (8 bits)

- **Character Constant**

Enclosed in single quotation mark

e.g. 'A' , 'c' , '2'

Character Variables & Constants

Contd...



```
//Demonstrates Character variables
#include<iostream>
int main()
{
    char ch1=88;
    char ch2='A'; // 'A' is character constant
    cout<<ch1<<"\n"<<ch2;
    ch1++;
    cout<<ch1;
    return 0;
}
```

Floating Point Types



Type	Bits	Range	Syntax
float	4 bytes	$3.4 \times 10^{-38} \rightarrow 3.4 \times 10^{38}$	float var;
double	8 bytes	$1.7 \times 10^{-308} \rightarrow 1.7 \times 10^{308}$	double var;
long double	16 bytes	$1.2 \times 10^{-4932} \rightarrow 1.2 \times 10^{4932}$	long double var;

Floating Point Types (Contd...)



```
//Floating point types
#include<iostream>
int main()
{
float f=312.4F; //type float constant with letter F
double d=2.34E5;
long double l=2.345E10L; //type float constant with letter L
cout<<“\n Type float:”<<f<<“\n Type double”<<d<<“\n Type
long double”<<l;
return 0;
}
```

Expression & Precedence



- Any combination of variables, constants and operators
 - $a+b$
 - $a/7$
 - $(a+b)*9$
- Parenthesis has higher precedence.
- $*$ and $/$ have same precedence.
- $+$ and $-$ have same precedence.
 - $20-3*7 \rightarrow 20-21 \rightarrow -1$
 - $(20-3)*7 \rightarrow 17*7 \rightarrow 119$
- Operators with same precedence are evaluated from left to right
 - $(15-5)*5/10 \rightarrow 10*5/10 \rightarrow 50/10 \rightarrow 5$ (left to right)

Input with cin



```
//Method to take user input
```

```
#include<iostream>
```

```
int main()
```

```
{
```

```
int num1, num2;
```

```
cout<<"Enter 1st number :";
```

```
cin>>num1;
```

```
cout<<"Enter 2nd number :";
```

```
cin>>num2;
```

or

```
cout<<"Enter two numbers :";
```

```
cin>>num1>>num2;
```

```
cout<<"\n You entered:"<<num1<<"&"<<num2;
```

```
return 0;
```

```
}
```

const Qualifier



```
//const Qualifier
#include<iostream>
int main()
{
int num1=15;
const int num2=23;
const float num3=3.14F;
const int num4; //ERROR
num1=20;
num2=50; //ERROR
return 0;
}
```

#define Directive



```
//#define Directive
#include<iostream>
#define PI 3.14
int main()
{
int num1;
num1=3*PI;
cout<<"Result="<<num1;
return 0;
}
```

Type bool



- 1 bit of storage...
- Only two values.. 0 and 1

```
//Type bool
#include<iostream>
int main()
{
bool b,b1;
b=5>3; //result is 1
b1=3>5; //result is 0
cout<<"b="<<b<<"b1="<<b1;
return 0;
}
```

setw Manipulator



```
//setw Manipulator
#include<iostream>
#include<iomanip>
int main()
{
int a=1234567;
int b=234;
int c=12345;
cout<<"a="<<setw(10)<<a;
cout<<"\nb="<<setw(10)<<b;
cout<<"\nc="<<setw(10)<<c;
return 0;
}
```

unsigned Data Types



Type	Low	High	Syntax
unsigned char	0	255	unsigned char var;
unsigned short	0	65,535	unsigned short var;
unsigned int	0	4,294,967,295	unsigned int var;
unsigned long	0	4,294,967,295	unsigned long var;

Type Conversion



- Automatic Conversion

Data Type	Order
long double	Highest
double	
float	
long	
int	
short	
char	Lowest

Automatic Conversion



```
//automatic conversion
#include<iostream>
int main()
{
int a=20;
float b=2.34F;
double result;
result=a*b; //automatic conversion
cout<<"Result="<<result;
return 0;
}
```

Arithmetic Operators



- Four arithmetic operators work on all data types.
 - **Addition (+)** `cout<<6+8;`
 - **Subtraction(-)** `float f=3.4-2.0;`
 - **Multiplication(*)** `long m=34*20;`
 - **Division(/)** `cout<<8/5;`

- **Remainder/Modulus Operator: (%)**

Works only with (*int, char, long, short*)

`cout<<6%8` → 6

`cout<<9%8` → 1

Arithmetic Assignment Operators



- Arithmetic assignment operators corresponding to all arithmetic operators are

$+=$, $-=$, $*=$, $/=$, $\%=$

$\text{item}+=5$ same as $\text{item}=\text{item} + 5$

$\text{item}-=6$ same as $\text{item}=\text{item} - 6$

$\text{item}* =4$ same as $\text{item}=\text{item} * 4$

$\text{item}/ =9$ same as $\text{item}=\text{item} / 9$

$\text{item}\% =7$ same as $\text{item}=\text{item} \% 7$

Increment Operator



- Increment operator `++` adds 1 to its argument

- **Prefix**

- `++count`

- **Postfix**

- `count++`

```
int count=10;
```

```
cout<<"count="<<count<<endl;
```

```
count=10
```

```
cout<<"count="<<++count<<endl;
```

```
count=11
```

```
cout<<"count="<<count<<endl;
```

```
count=11
```

```
cout<<"count="<<count++<<endl;
```

```
count=11
```

```
cout<<"count="<<count<<endl;
```

```
count=12
```

Decrement Operator



- Decrement operator -- subtracts 1 from its argument
 - **Prefix**
--count
 - **Postfix**
count--

```
int count=10;
```

```
cout<<"count="<<count<<endl;
```

```
cout<<"count="<<--count<<endl;
```

```
cout<<"count="<<count<<endl;
```

```
cout<<"count="<<count--<<endl;
```

```
cout<<"count="<<count<<endl;
```

```
count=10
```

```
count=9
```

```
count=9
```

```
count=9
```

```
count=8
```