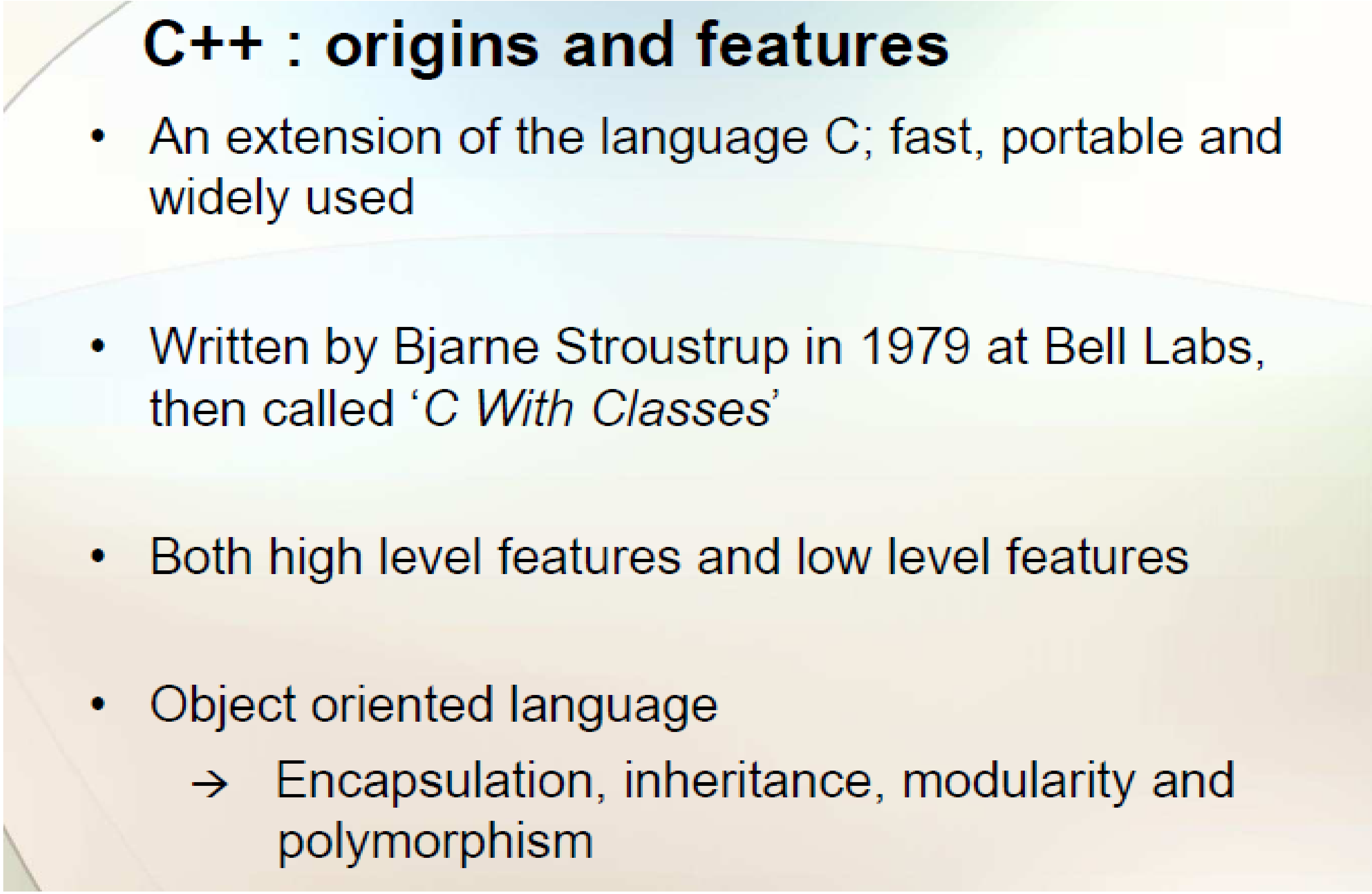
CSC 201 CSC 150
C++ Programming

Dr. Mazen Zainedin

Stat & OR Dept.

College of sciences KSU

Lecture 1: Introduction to C++



C++ : origins and features

- An extension of the language C; fast, portable and widely used
- Written by Bjarne Stroustrup in 1979 at Bell Labs, then called '*C With Classes*'
- Both high level features and low level features
- Object oriented language
  - → Encapsulation, inheritance, modularity and polymorphism

The compilation process

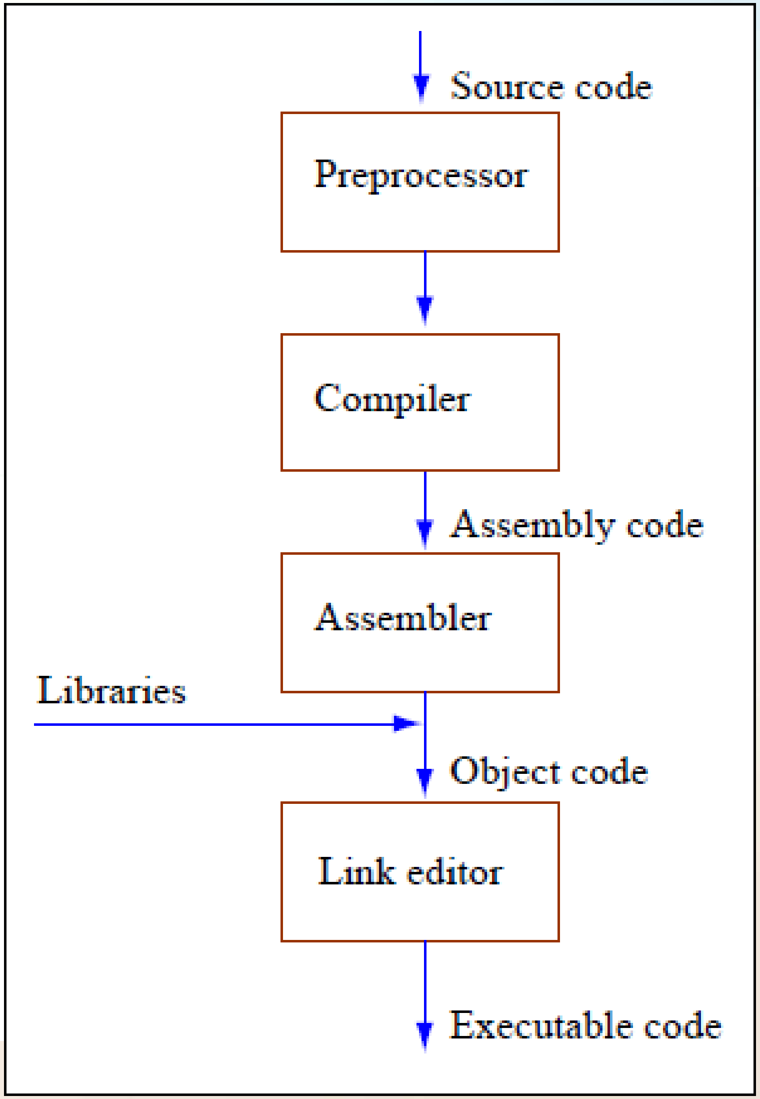


Figure by MIT OpenCourseWare.

- The preprocessor removes comments and handles directives for source file inclusions, definitions etc.

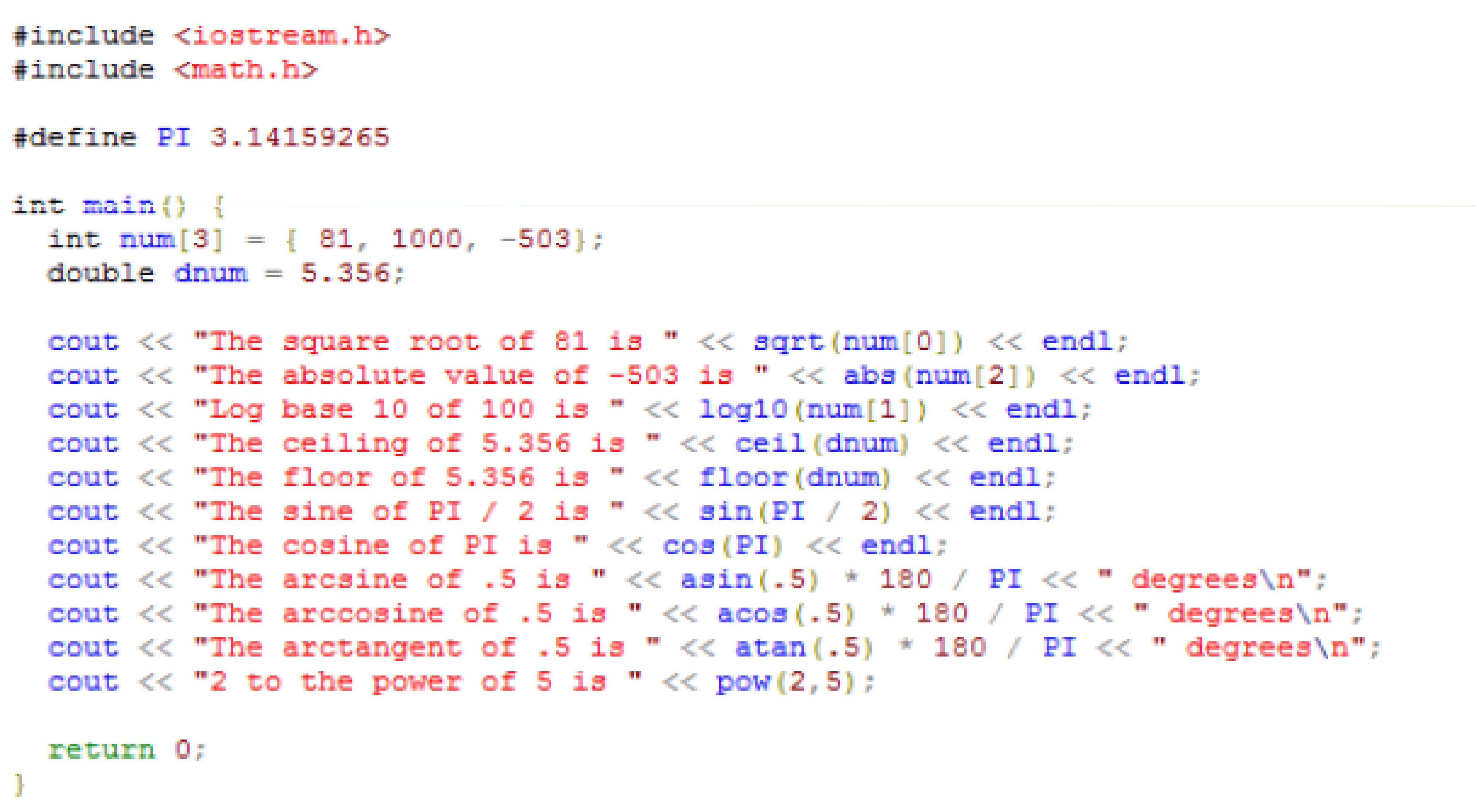
- The compiler translates source to assembly code

- The assembler creates object code, seen as a .OBJ file.

- The link editor combines any referenced functions from the libraries and makes an executable code.

Header files

- Store the functions in the standard C++ library
- iostream.h, conio.h, stdio.h, math.h etc



```
#include <iostream.h>
#include <math.h>

#define PI 3.14159265

int main() {
  int num[3] = { 81, 1000, -503};
  double dnum = 5.356;

  cout << "The square root of 81 is " << sqrt(num[0]) << endl;
  cout << "The absolute value of -503 is " << abs(num[2]) << endl;
  cout << "Log base 10 of 100 is " << log10(num[1]) << endl;
  cout << "The ceiling of 5.356 is " << ceil(dnum) << endl;
  cout << "The floor of 5.356 is " << floor(dnum) << endl;
  cout << "The sine of PI / 2 is " << sin(PI / 2) << endl;
  cout << "The cosine of PI is " << cos(PI) << endl;
  cout << "The arcsine of .5 is " << asin(.5) * 180 / PI << " degrees\n";
  cout << "The arccosine of .5 is " << acos(.5) * 180 / PI << " degrees\n";
  cout << "The arctangent of .5 is " << atan(.5) * 180 / PI << " degrees\n";
  cout << "2 to the power of 5 is " << pow(2,5);

  return 0;
}
```

A HelloWorld program

-
  ```
  // First program
  #include <iostream>
  using namespace std;
  int main ()
  {
          cout << "Hello World!";
         return 0;
  }
  ```

Output:

```
Hello World!
```

A HelloWorld program

- // - Comments: no effect on the behavior of the program
- *#include <iostream>* tells the preprocessor to include the iostream standard file
- *using namespace std;* uses all the elements of the standard C++ library, declared within what is called a namespace, the namespace with the name *std*.
- *int main ()* – starts the main function
- *cout* << *"Hello World!";* - prints the string
- *return 0;* - causes the main to finish

The main() function

- Execution of the program starts here, the first thing done

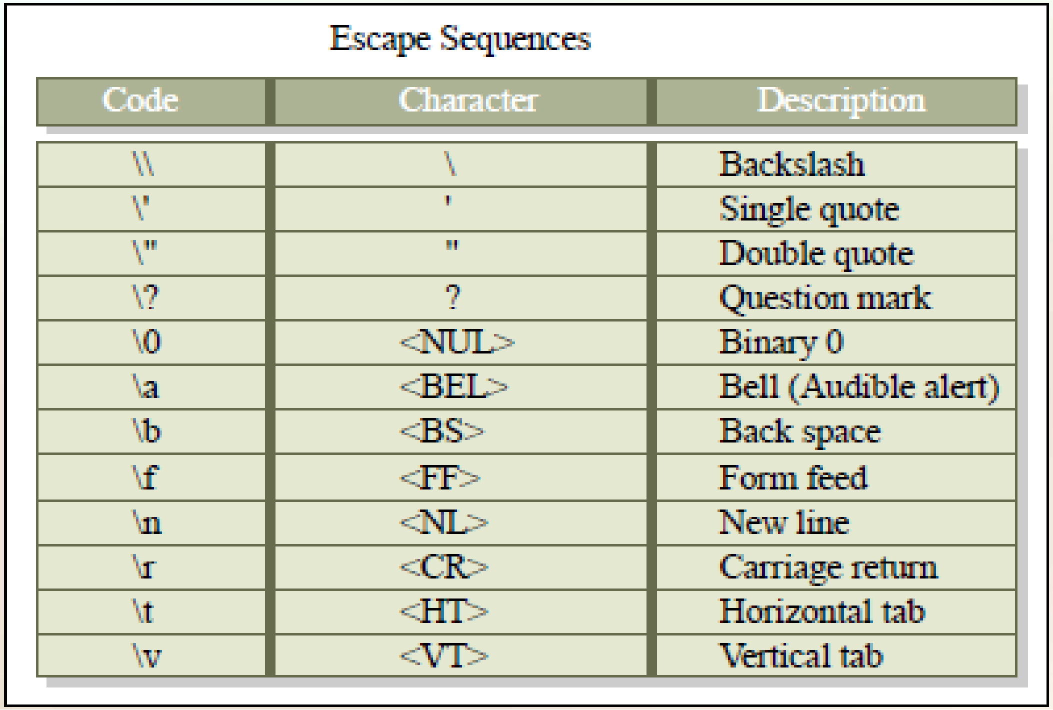
- The return type of *main* must be int

- *main* may not be called from inside the program, only the system may call the *main* function.

- int main() { }

C++ character set

- 0 1 2 3 4 5 6 7 8 9
- A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
- a b c d e f g h i j k l m n o p q r s t u v w x y z
- _ $ { } [ ] # ( ) < > % : ; . ? * + - / ^ & | ~ ! = , \ " ‘ <SPACE>



Escape Sequences

Code	Character	Description
\\	\	Backslash
\'	'	Single quote
\"	"	Double quote
\?	?	Question mark
\0	<NUL>	Binary 0
\a	<BEL>	Bell (Audible alert)
\b	<BS>	Back space
\f	<FF>	Form feed
\n	<NL>	New line
\r	<CR>	Carriage return
\t	<HT>	Horizontal tab
\v	<VT>	Vertical tab

C++ Tokens

- A token is the smallest element of a C++ program that is meaningful to the compiler.

- Kinds of tokens: identifiers, keywords, literals, operators, punctuators, and other separators.

- Tokens are usually separated by "white space." White space can be one or more blanks, horizontal or vertical tabs, new lines, formfeeds or comments.

C++ Keywords

- This is a list of reserved keywords in C++. Since they are used by the language, these keywords are not available for use by programmers.

- auto const double float int short struct
  unsigned break continue else for long signed
  switch void case default enum goto register
  sizeof typedef volatile char do extern if
  return static union while ,**ETC.**

- You cannot define classes, variables, or functions that have these keywords as their names.

Assignment operators

- The assignment operator (=) causes the operand on the ***left*** side of the assignment operator to have its value changed to the value on the ***right*** side of the assignment operator.

- x = a + b;
  x = 35;

- 35 = x;   // Wrong

Mathematical Operators

- Addition (+), subtraction (-), multiplication (*), division (/), and modulus (%)

- Integer Division : 21 / 5 = 4

- Modulus : Gives the remainder. 21 % 4 = 1

-
  ```
  int myAge = 5;
  int temp;
  temp = myAge + 2; //add 5 + 2 and put it in temp
  myAge = temp;    // put it back in myAge
  ```

Unary and binary operators

- Binary operators : act on two operands

  (+) a+b,   (%) a%b

- Unary operators : act on one operand only

  ! (Logical NOT)   !a

  To nullify a variable, you can write the exclamation point to its left.

- & , ~ , * , + , ++ , – , —

Separators

- Types of separators:

  { } -- often used for function definitions

  ( ) -- often used in conditional statements

  , -- comma

  ; -- used for ending a syntax

- [ ] :

Data types in C++

- What kind of data? An integer? A decimal? A letter?

Name	Description	Size	Range
char	Character / small integer	1byte	signed: -128 to 127 unsigned: 0 to 255
short int (short)	Short Integer.	2bytes	signed: 32768 to 32767 unsigned: 0 to 65535
int	Integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
long int (long)	Long integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295

Data types in C++

Name	Description	Size	Range
bool	Boolean value. It can take one of two values: true or false.	1bit	true or false (1 or 0)
float	Floating point number.	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	Double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	Long double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)

Variable declaration and naming conventions

- Write the data type first, then the variable name:

```
int a;
char ch = ‘a’ ;
float x , y= 5.2 ;  //Separated by a comma
```

- Clarity and readability. Relevance. Verbs for functions.

```
line, savingsAccount , getName()
```

- Constants are all in uppercase.

```
float PI = 3.141592653;
```

Starting to write programs

- Include header files

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

- Declare the *main* function

```
int main () {
```

- Declare your variables

```
int a, b; float c;
```

Starting to write programs

- Printing to the screen:

```
  cout<<“Hello”;
 cout<<“I like C++ ” << “more than Java.”;
```

- Taking an input:

```
int a;                           // declare the variable
cout<<“Enter the value of a”; //Display a message
cout<<endl;
 cin>>a;
```

- Write the rest of the code and close all brackets.