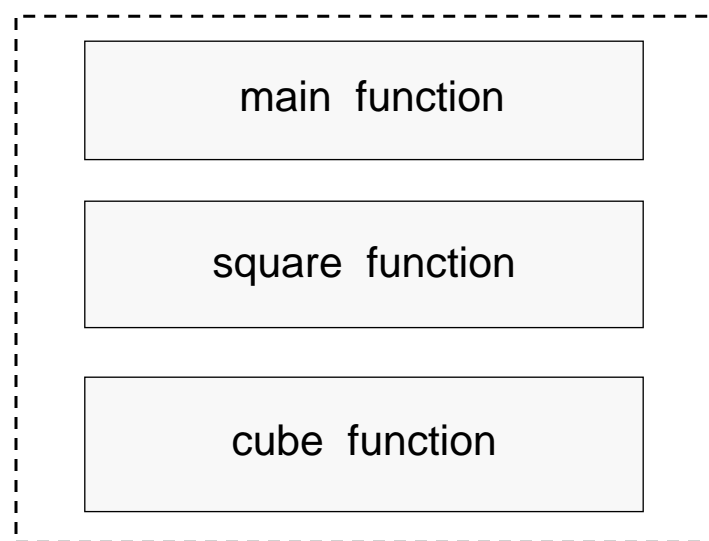


## A C++ program is a collection of one or more functions

- there must be a function called main( )
- execution always begins with the first statement in function main( )
- any other functions in your program are subprograms and are not executed until they are called

3

## Program With Several Functions



4

## Program With Three Functions

```
#include <iostream>

int Square( int );      // declares these two
int Cube( int );       // value-returning functions

using namespace std ;

int main( )
{
    cout << "The square of 27 is "
          << Square(27) << endl;      // function call
    cout << "The cube of 27 is "
          << Cube(27) << endl;      // function call
    return 0;
}
```

5

## Rest of Program

```
int Square( int n )
{
    return n * n;
}

int Cube( int n )
{
    return n * n * n;
}
```

6

## Output of program

```
The square of 27 is 729  
The cube of 27 is 19683
```

7

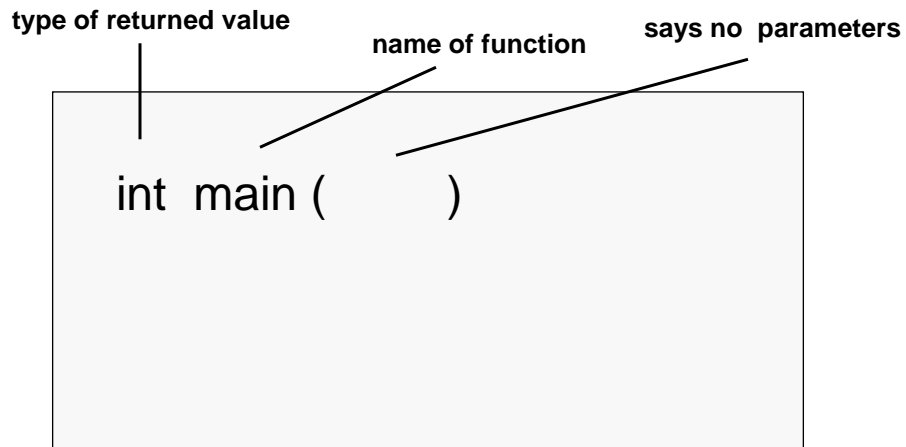
## Shortest C++ Program

type of returned value      name of function

```
int main ( )  
{  
  
    return 0;  
}
```

8

## What is in a heading?

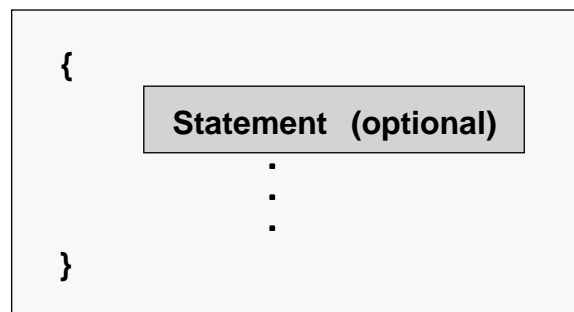


9

## Block (Compound Statement)

- a block is a sequence of zero or more statements enclosed by a pair of curly braces  
{ }

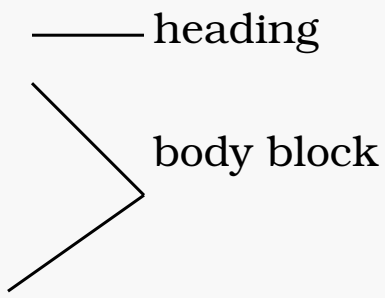
### SYNTAX



10

## Every C++ function has 2 parts

```
int main ( )  
{  
  
    return 0;  
}
```



The diagram illustrates the two parts of a C++ function. A horizontal line points to the text "int main ( )" and is labeled "heading". A large right-pointing chevron shape encompasses the code block between the opening and closing curly braces, and is labeled "body block".

11

## What is an Identifier?

- An identifier is the name used for a data object (a variable or a constant), or for a function, in a C++ program.
- C++ is a case-sensitive language.
- using meaningful identifiers is a good programming practice

17

## Identifiers

- an identifier must start with a letter or underscore, and be followed by zero or more letters (A-Z, a-z), digits (0-9), or underscores

- **VALID**

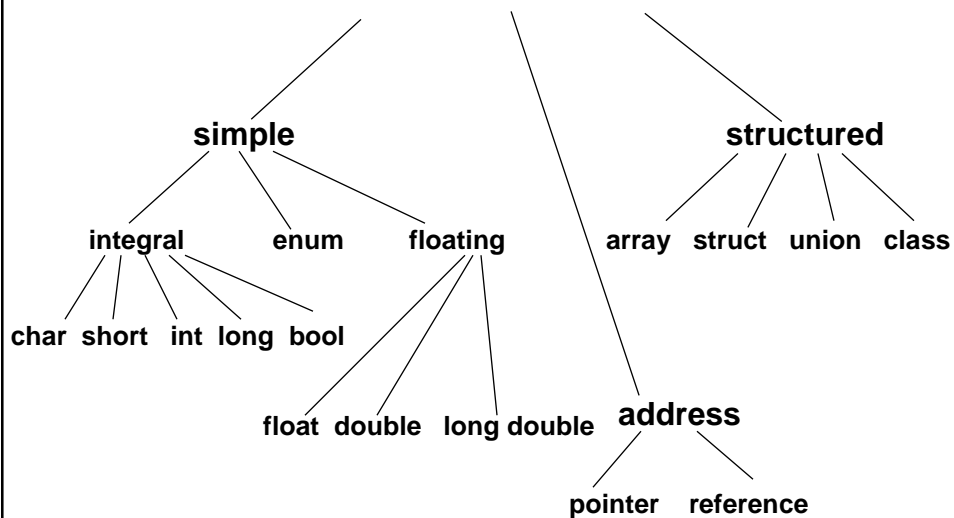
age_of_dog	taxRateY2K
PrintHeading	ageOfHorse

- **NOT VALID (Why?)**

age#	2000TaxRate	Age-Of-Cat
------	-------------	------------

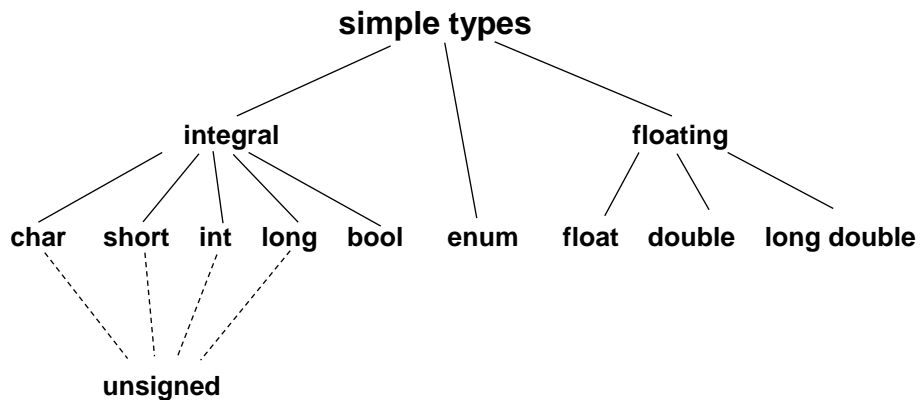
18

## C++ Data Types



22

## C++ Simple Data Types



23

## C++ Data Type String

- a string is a sequence of characters enclosed in double quotes
- string sample values  
"Hello"    "Year 2000"    "1234"
- the empty string (null string) contains no characters and is written as ""

25

## More About Type String

- **string is not a built-in (standard) type**
  - it is a programmer-defined data type
  - it is provided in the C++ standard library
- **string operations include**
  - comparing 2 string values
  - searching a string for a particular character
  - joining one string to another

26

## What is a Variable?

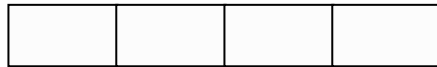
- A variable is a location in memory which we can refer to by an identifier, and in which a data value that can be changed is stored.
- declaring a variable means specifying both its name and its data type

28

## What Does a Variable Declaration Do?

```
int    ageOfDog;
float  taxRateY2K;
char   middleInitial;
```

A declaration tells the compiler to allocate enough memory to hold a value of this data type, and to associate the identifier with this location.



4 bytes for taxRateY2K



1 byte for middleInitial

29

## What is a Named Constant?

- A named constant is a location in memory that we can refer to by an identifier, and in which a data value that cannot be changed is stored.

### VALID CONSTANT DECLARATIONS

```
const string STARS = "*****";
const float  NORMAL_TEMP = 98.6;
const char   BLANK = ' ';
const int    VOTING_AGE = 18;
const float  MAX_HOURS = 40.0;
```

33