Data Types

- Data Type defines data representation
  - Range of values that can be stored
  - Determines operations allowed
- C++ utilizes *strict data typing* of variables
  - 4 Basic Data Types: char int float double
  - 4 Type Modifiers: signed unsigned long short
  - 2 New Basic Data Types:
    - bool may contain only true or false values
    - wchar_t international characters = unicode
- sizeof(*variable*) Operator
  - Returns a value which describes the total number of bytes utilized by a variable for storage

Character Data Type Declaration

- char

  Reserves 8 bits (1 byte) of RAM memory which can represent:
  - ASCII character 'A' 'a' '1' '4' '*' '?'
  - Signed integer in range 127 to -128 (Default)
  - Unsigned integer in range 255 to 0
  - Examples:

```
char cLetter;
char cGradePoints, cGrade;
unsigned char cWeekNumber = 200;
char cGradeA = 65, cGradeB = 'B';
```

```
Attendance Grade = A
Attendance Score = C
Size of cGrade = 1
Enter a Grade: G
You entered G
Press any key to continue . . .
```

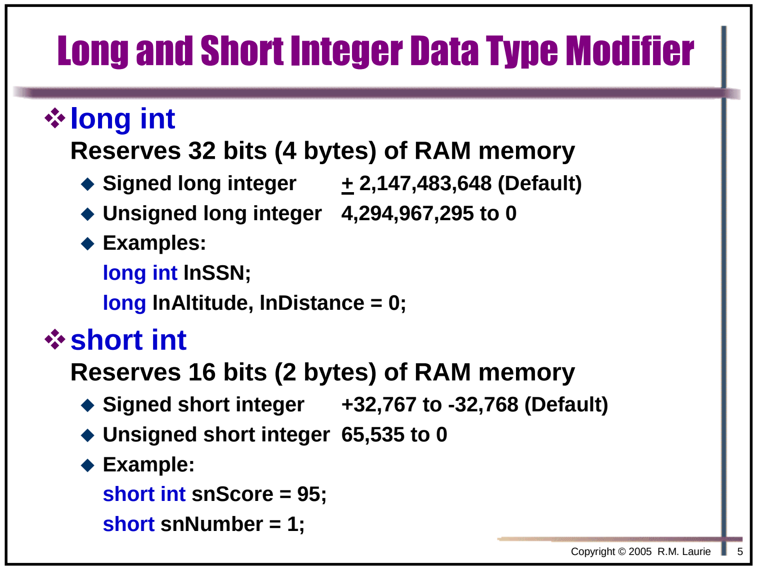
```
/**************************
* PROGRAM: CharType.cpp
**************************
#include <iostream>
using namespace std;
int main()
{
  char  cGrade = 'A', cScore = 67;
  int   nSize;
  nSize = sizeof(cGrade);
  cout  << "Attendance Grade = " << cGrade
        << endl
        << "Attendance Score = " << cScore
        << endl
        << "Size of cGrade = " << nSize << endl;
  cout  << "Enter a Grade: ";

  cin   >> cGrade;
  cout  << "You entered" << cGrade << endl;
  system("pause");
  return 0;
}
```

Integer Data Type Declarations

- int

  Reserves one *word* of RAM memory which can represent:
  - 16 bits (2 bytes) for Win16
    - Signed integer 32,767 to -32,768 (Default)
    - Unsigned integer 65,535 to 0
  - 32 bits (4 bytes) for Win32 (Borland C++)
    - Signed integer ± 2,147,483,647 (Default)
    - Unsigned integer 4,294,967,295 to 0
  - Examples:

    int nSSN = 390546348;
    int nTotalScore, nClassMedian;
    unsigned int unHeight = 100, unWidth = 50000;



Long and Short Integer Data Type Modifier

- **long int**
  - **Reserves 32 bits (4 bytes) of RAM memory**
    - **Signed long integer** **$\underline{+}$ 2,147,483,648 (Default)**
    - **Unsigned long integer** **4,294,967,295 to 0**
    - **Examples:**

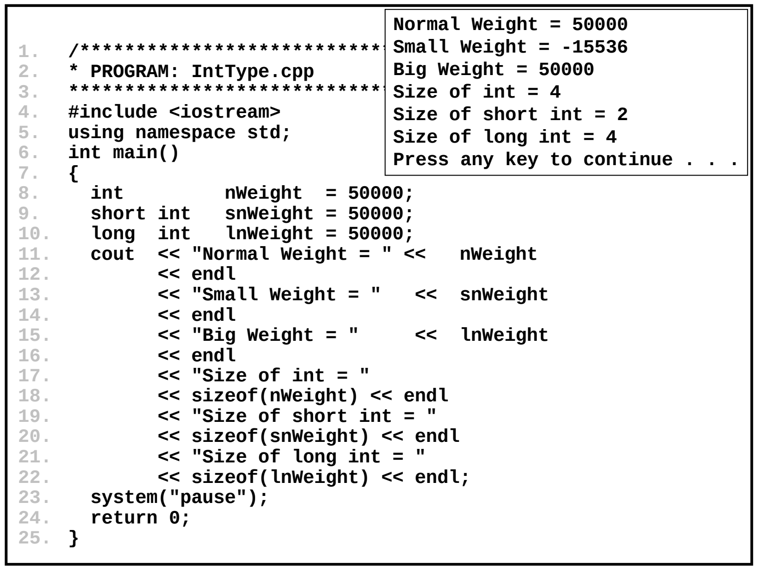
      long int lnSSN;

      long lnAltitude, lnDistance = 0;
- **short int**
  - **Reserves 16 bits (2 bytes) of RAM memory**
    - **Signed short integer** **+32,767 to -32,768 (Default)**
    - **Unsigned short integer** **65,535 to 0**
    - **Example:**

      short int snScore = 95;

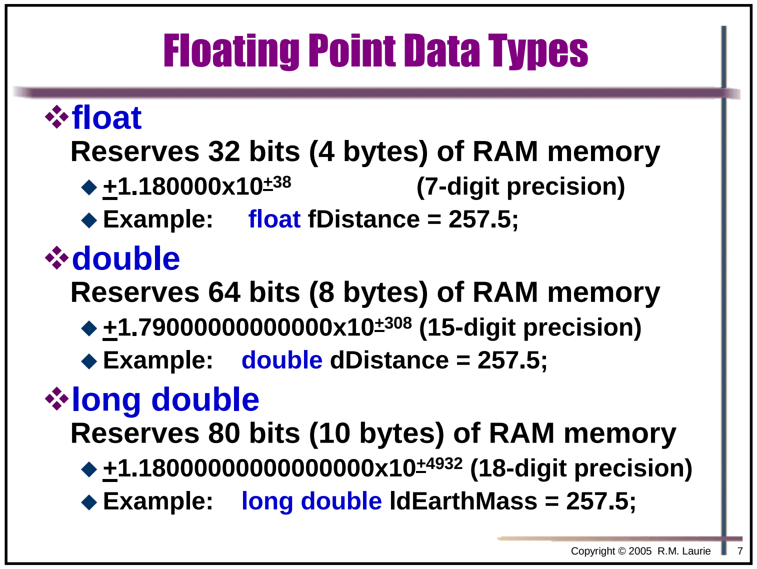
      short snNumber = 1;

5



```
/*****************************
* PROGRAM: IntType.cpp
*****************************
#include <iostream>
using namespace std;
int main()
{
  int         nWeight  = 50000;
  short int   snWeight = 50000;
  long  int   lnWeight = 50000;
  cout  << "Normal Weight = " <<   nWeight
        << endl
        << "Small Weight = "   <<  snWeight
        << endl
        << "Big Weight = "     <<  lnWeight
        << endl
        << "Size of int = "
        << sizeof(nWeight) << endl
        << "Size of short int = "
        << sizeof(snWeight) << endl
        << "Size of long int = "
        << sizeof(lnWeight) << endl;
  system("pause");
  return 0;
}
```

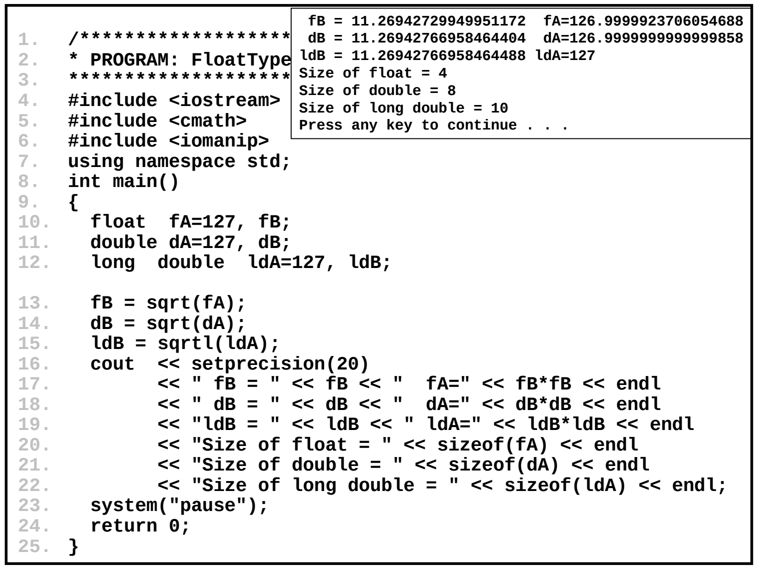
```
Normal Weight = 50000
Small Weight = -15536
Big Weight = 50000
Size of int = 4
Size of short int = 2
Size of long int = 4
Press any key to continue . . .
```



Floating Point Data Types

- **float**
  - **Reserves 32 bits (4 bytes) of RAM memory**
    - **$\underline{+}1.180000\text{x}10^{\pm38}$ (7-digit precision)**
    - **Example: float fDistance = 257.5;**
- **double**
  - **Reserves 64 bits (8 bytes) of RAM memory**
    - **$\underline{+}1.790000000000000\text{x}10^{\pm308}$ (15-digit precision)**
    - **Example: double dDistance = 257.5;**
- **long double**
  - **Reserves 80 bits (10 bytes) of RAM memory**
    - **$\underline{+}1.180000000000000000\text{x}10^{\pm4932}$ (18-digit precision)**
    - **Example: long double ldEarthMass = 257.5;**

7



```
/********************
* PROGRAM: FloatType
********************
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;
int main()
{
  float  fA=127, fB;
  double dA=127, dB;
  long  double  ldA=127, ldB;

  fB = sqrt(fA);
  dB = sqrt(dA);
  ldB = sqrtl(ldA);
  cout  << setprecision(20)
        << " fB = " << fB << "  fA=" << fB*fB << endl
        << " dB = " << dB << "  dA=" << dB*dB << endl
        << "ldB = " << ldB << " ldA=" << ldB*ldB << endl
        << "Size of float = " << sizeof(fA) << endl
        << "Size of double = " << sizeof(dA) << endl
        << "Size of long double = " << sizeof(ldA) << endl;
  system("pause");
  return 0;
}
```

```
 fB = 11.269427299499511172  fA=126.9999923706054688
 dB = 11.269427669584644040  dA=126.9999999999999858
ldB = 11.269427669584644880 ldA=127
Size of float = 4
Size of double = 8
Size of long double = 10
Press any key to continue . . .
```

Boolean Data Type

- **bool**
  Reserves 1 bit of RAM memory
  - Generally, 1 byte because smallest addressable
  - Evaluates as true/false
  - Logical operators can be applied:
    - && Logical AND Operator
    - | | Logical OR Operator
    - ! Logical NOT Operator

bool bWorkDay = true, bRainDay = true, bGo2Work;
bGo2Work = bWork && !bRadDay;

```
/**********************************
* PROGRAM: BoolType.cpp
**********************************
#include <iostream>
#include <string>
using namespace std;
int main()
{
  bool bWorkDay = true, bRainDay = false, bGo2Work;
  bGo2Work = bWorkDay && !bRainDay;
  cout  << "Work day = " << bWorkDay << endl
        << "Rain day = " << bRainDay << endl
        << "----------------------" << endl
        << "Go work  = " << bGo2Work << endl << endl;
  bRainDay = true,
  bGo2Work = bWorkDay && !bRainDay;
  cout  << "Work day = " << bWorkDay << endl
        << "Rain day = " << bRainDay << endl
        << "----------------------" << endl
        << "Go work  = " << bGo2Work << endl << endl;
  system("pause");
  return 0;
}
```

```
Work day = 1
Rain day = 0
----------------------
Go work  = 1

Work day = 1
Rain day = 1
----------------------
Go work  = 0
```

Variable Identifier Naming Conventions

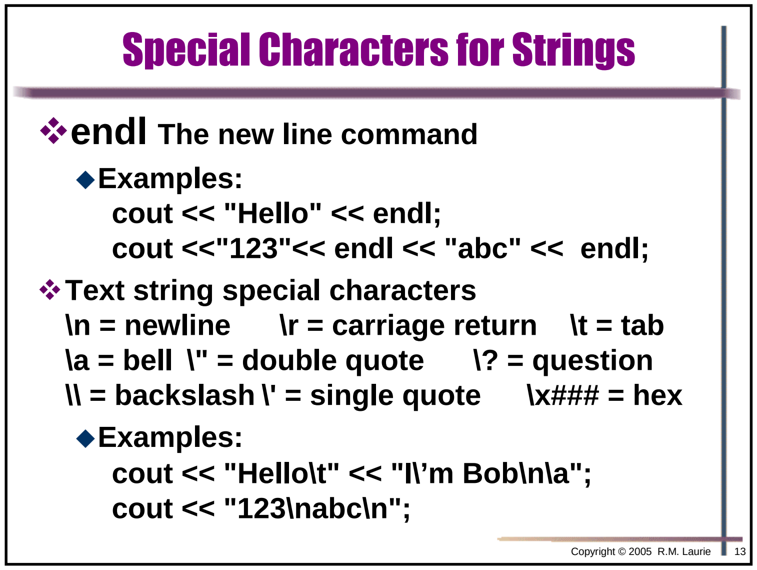
- **Variable Identifier begins with lower case letter(s) to indicate data type**

PREFIX	DATA TYPE (Bit Length for Borland C++)
c	Char (8 bits)
n	Integer (1 word = 16 bits or 32 bits)
un	Unsigned Integer
sn	Short Integer (16 bits)
ln	Long Integer (32 bits)
f	Float (32 bits)
d	Double (64 bits)
ld	Long Double (80 bits)
b	Boolean (1 bit → 1 byte min addressable)

DataType Literals

- **Literals are fixed human-readable values that can not be altered by program**

LITERALS	DATA TYPE
'A'	Char
"Hello"	String of characters
+3 12 -123	Integer
40000U	Unsigned Integer
35000L	Long Integer
35000UL	Unsigned Long Integer
123.45F -4.1E-2F	Float
123.45 -4.1E-2	Double
123.45L -4.1E-2L	Long Double
0x4F 0x6B 0x21	Hexadecimal (Base 16)
026 001	Octal (Base 8)



Special Characters for Strings

- **endl** The new line command
  - Examples:

    cout << "Hello" << endl;

    cout <<"123"<< endl << "abc" << endl;
- Text string special characters

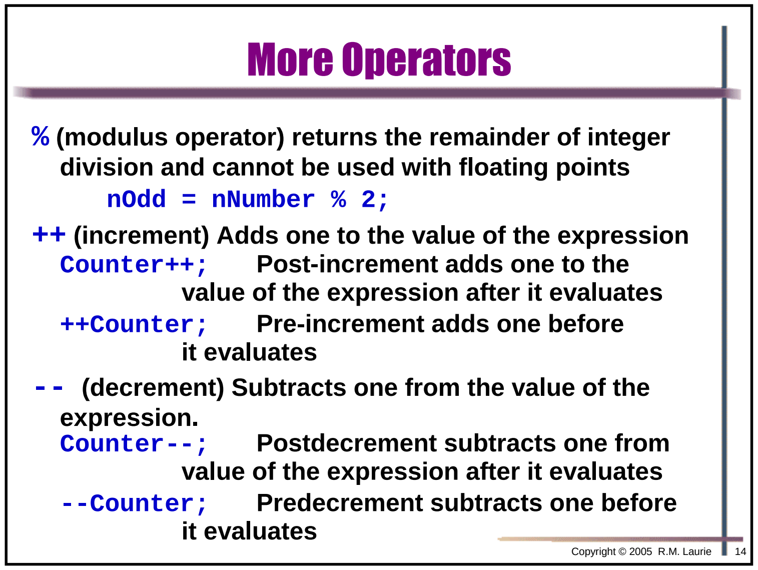
  \n = newline   \r = carriage return   \t = tab

  \a = bell \" = double quote   \? = question

  \\ = backslash \' = single quote   \x### = hex
  - Examples:

    cout << "Hello\t" << "I\'m Bob\n\a";

    cout << "123\nabc\n";



More Operators

% (modulus operator) returns the remainder of integer division and cannot be used with floating points

```
nOdd = nNumber % 2;
```

++ (increment) Adds one to the value of the expression

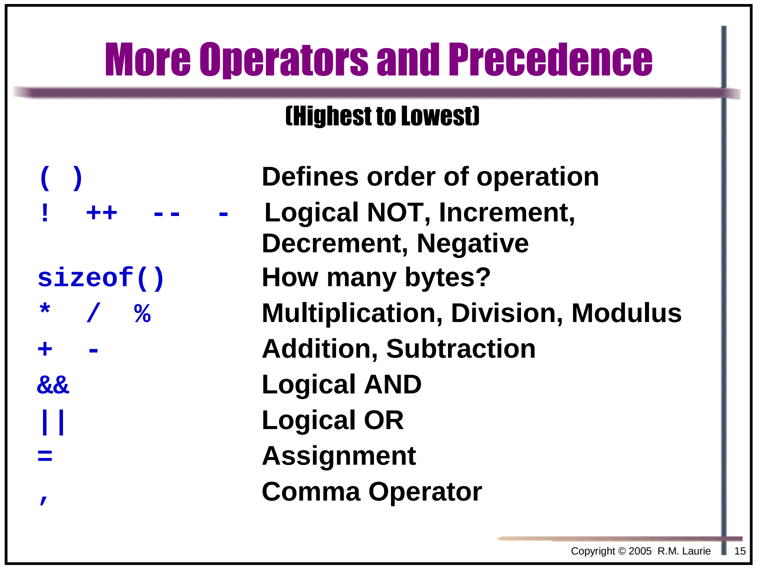
`Counter++;` Post-increment adds one to the value of the expression after it evaluates

`++Counter;` Pre-increment adds one before it evaluates

-- (decrement) Subtracts one from the value of the expression.

`Counter--;` Postdecrement subtracts one from value of the expression after it evaluates

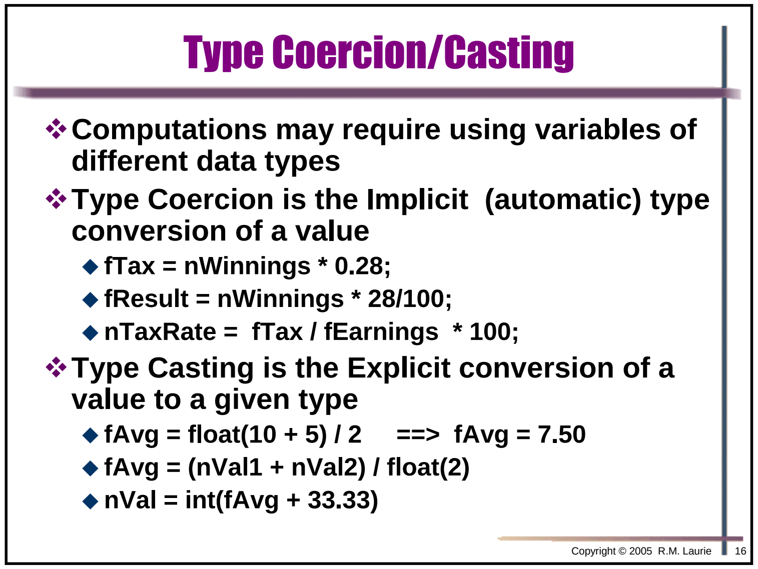
`--Counter;` Predecrement subtracts one before it evaluates



More Operators and Precedence

(Highest to Lowest)

`( )`	Defines order of operation
`! ++ -- -`	Logical NOT, Increment, Decrement, Negative
`sizeof()`	How many bytes?
`* / %`	Multiplication, Division, Modulus
`+ -`	Addition, Subtraction
`&&`	Logical AND
`||`	Logical OR
`=`	Assignment
`,`	Comma Operator



Type Coercion/Casting

- Computations may require using variables of different data types
- Type Coercion is the Implicit (automatic) type conversion of a value
  - fTax = nWinnings * 0.28;
  - fResult = nWinnings * 28/100;
  - nTaxRate = fTax / fEarnings * 100;
- Type Casting is the Explicit conversion of a value to a given type
  - fAvg = float(10 + 5) / 2 ==> fAvg = 7.50
  - fAvg = (nVal1 + nVal2) / float(2)
  - nVal = int(fAvg + 33.33)