Chapter 3

Tuesday, August 22, 2017 1:55 PM

Chapter 3: Expressions and Interactivity

3.1 The `cin` Object

- To read input from the user
- Requires `include <iostream>`

```
#include <iostream>
using namespace std;
int main(){
      double height, weight;
      cout << "Enter height and weight separated by a space:\n";

      //to just read in height, use cin >> height;

      cin >> height >> weight;
      cout << "Height is: " << height << endl;
      cout << "Weight is: " << weight << endl;
      system("pause");
      return 0;
}
```

3.2 Mathematical Expressions

- Combines Constants and Variables to calculate new values

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

int main() {
    double height, weight;
    cout << "Enter height and weight separated by a space:\n";

    //to just read in height, use cin >> height;

    cin >> height >> weight;
    const double BMI = (703 * weight) / (height * height);
```

```
    cin >> height >> weight;
    const double BMI = (703 * weight) / (height * height);
    cout << "Height is: " << height << endl;
    cout << "Weight is: " << weight << endl;
    cout << "BMI: " << BMI << endl;
    system("pause");
    return 0;
}
```

PEMDAS
Parentheses
Exponents
Multiplication
Division
Addition
Subtraction

2 + 2 * 2 - 2
2 + 4 - 2
6 - 2
4

(2 + 2) * 2 - 2
4 * 2 - 2
8 - 2
6

(2 + 2) * (2 - 2)
4 * (2 - 2)
4 * 0
0

3.3 When You Mix Apples with Oranges: Type Conversion

- When operating on values of different data types, the lower one is promoted to the type of the higher one.

- When using the = operator, the type of the expression on the right will be converted to the variable type on the left.

  float total = 99.7;
  int n = 5;
  double result = total / n;

- result will be of data type double
- 19.9 is stored in result

  float a = 99.7;
  int b = 5;
  int c = a / b;

- c will be of data type int

int c = a / b;

- c will be of data type int
- 19 is stored in c

3.4 Overflow and Underflow

- Occurs when assigning a value that is too large (overflow) or too small (underflow) to be held in a variable

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

int main() {
     int tooBig = 100 * 100 * 100 * 100;
     cout << tooBig << endl;

     system("pause");
     return 0;
}
```

3.5 Type Casting

- Used for manual data type conversion
- Useful for floating point division using ints
- Useful to see int value of a char variable

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

int main() {
     cout << "How many books do you plan to read?" << endl;
     int books;
     cin >> books;

     cout << "How many months will it take?" << endl;
     int months;
     cin >> months;

     double perMonth = static_cast<double>(books) / months;

     cout << "That is " << perMonth << " books per month\n";
     system("pause");
     return 0;
}
```

3.6 Multiple Assignment and Combined Assignment

Multiple assignment - assigning values to multiple variables with =

x = y = z = 5;

Combined assignment - changing the value of a variable and storing the result in the original variable

Expression	Equivalent to (Combined Assignment Operators)	Equivalent to
sum = sum + 1;	sum += 1;	sum++;
sum = sum - 1;	sum -=1;	sum--;
sum = sum * 2;	sum *= 2;	
sum = sum / 2;	sum /= 2;	
sum = sum % 2;	sum %= 2;	

Assume x = 6

Combined Assignment Operator	New Value of x
x += 4	10
x -= 3	3
x *= 10	60
x /= 2	3
x %= 4	2

3.7 Formatting Output

- Requires <iomanip> header file
- #include <iomanip>

Stream Manipulators

setw(x)	print in a field at least x spaces wide
fixed	displays the output in decimal format rather than scientific notation
showpoint	forces all floating-point output to show a decimal point, even if the values are whole numbers
setprecision(y)	Rounds all floating-point numbers to y decimal places

- cout << setw(15) << 1234 << endl;

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

int main() {
	int height, weight;
	cout << "Enter height and weight separated by a space:\n";
```

```
int main() {
	int height, weight;
	cout << "Enter height and weight separated by a space:\n";

	//to just read in height, use cin >> height;

	cin >> height >> weight;
	const double BMI = (703 * weight) / (height * height);
	cout << fixed << showpoint << setprecision(2);
	cout << "Height is: " << setw(15) << height << endl;
	cout << "Weight is: " << setw(15) << weight << endl;
	cout << setw(11) << "BMI: " << setw(15) << BMI << endl;
	system("pause");
	return 0;
}
```

3.8 Working with Characters and string Objects

string	series of characters

- #include<string>
- `string s = "New York";`
- `string name = "Rohini";`
- `string message = "Happy Birthday, ";`

Member Functions and Operators

length()	`s.length()`	8		
+ (concatenation)	`message + name`	`Happy Birthday, Rohini`	`"Hi " + name`	`Hi Rohini`

Reading in a line of text as a string using getline():

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

int main(){
    string name;
    string city;
    cout << "Please enter your name: ";
    //cin >> name;
    getline(cin, name);
    cout << "Please enter your city: ";
    getline(cin, city);

    cout << "Hello " << name <<endl;
    cout << "You live in " << city <<endl;

    system("pause");
    return 0;
```

```
    system("pause");
    return 0;
}
```

- `cin.get(x)` same as `cin >> x` but allows whitespace

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

int main(){
    char ch;
    cout << "Please enter a character, and I'll tell you the ASCII value: " <<endl;
    //cin >> ch;
    cin.get(ch);
    int asciiValue = static_cast<int>(ch);
    cout << asciiValue << endl;

    system("pause");
    return 0;
}
```

```
'A' --> 65
'B' --> 66
'W' --> 87
….
'a' --> 97
'v' --> 112
'w' --> 113
'x' --> 114
'?' --> 63
',' --> 44

..
'1' --> 49
```

cin.ignore()	Skips next char
cin.ignore(10, '\n')	Skips the next 10 chars or Until a '\n'

3.9 More Mathematical Library Functions

- Require cmath header file
- `#include <cmath>`

sin(x)	Sine of x
cos(x)	Cosine of x
tan(x)	Tangent of x

sqrt(x)	Square root of x
log(x)	Natural log (base e) of x
pow(x,y)	x^y

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

int main(){
    int number;
    cout << "What is your number?" << endl;
    cin >> number;
    cout << sqrt(number) << endl;

    System("pause");
    return 0;
}
```

3.10 Hand Tracing a Program

- Acting if you are the computer executing a program:
    - Step through and 'execute' each statement one by one
    - Record the contents of each variable after statement execution
        - Using a hand trace chart

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

int main(){
    int length, width, area;

    cout << "This program calculates the area of a ";
    cout << "rectangle.\n";
    cout << "What is the length of the rectangle? ";
    cin >> length;
    cout << "What is the width of the rectangle? ";
    cin >> width;
    area = length * width;
    cout << "The area of the rectangle is " << area << ".\n";
    return 0;
}
```

	length	width	area
int length, width, area;			
cout << "This program calculates the area of a ";			
cout << "What is the length of the rectangle? ";			
cin >> length;	10		
cout << "What is the width of the rectangle? ";	10		
cin >> width;	10	20	
area = length * width;	10	20	200

`cout << "The area of the rectangle is " << area << ".\n";`	10	20	200
`System("pause");`	10	20	200
`return 0;`			

3.11 A Case Study

- General Crates, Inc. builds custom-designed wooden crates
- You have been asked to write a program that
  - Calculates the
    - Volume (in cubic feet)
    - Cost
    - Customer price
    - Profit of any crate GCI builds

- Variables and Constants

COST_PER_CUBIC_FOOT	.23
CHARGE_PER_CUBIC_FOOT	.5
length	
width	
height	
volume	
cost	
charge	
profit	

- Step 1:
  - Ask the user to enter the dimensions of the crate
- Step 2:
  - Calculate:
    - The crate's volume
    - The cost of building the crate
    - The customer's charge
    - The profit made
- Step 3:
  - Display the data calculated in Step 2

Pseudocode:

Ask the user what the length is
Get input for length
Ask the user what the width is
Get input for width
Ask the user what the height is
Get input for height

*Volume = length * width * height*
Display crate's volume
*Cost = COST_PER_CUBIC_FOOT * volume*
Display crate's cost
*Charge = CHARGE_PER_CUBIC_FOOT * volume*
Display the charge
profit = charge – cost
Display the profit

Finished Code (For this, the order of the calculations and output are a little different: our pseudocode from class does each calculation and displays it right after, while the code below does all the calculations then displays them all.)

```
// This program is used by General Crates, Inc. to calculate
// the volume, cost, customer charge, and profit of a crate
// of any size. It calculates this data from user input, which
// consists of the dimensions of the crate.
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
   // Constants for cost and amount charged
   const double COST_PER_CUBIC_FOOT = 0.23;
   const double CHARGE_PER_CUBIC_FOOT = 0.5;

   // Variables
   double length,   // The crate's length
          width,    // The crate's width
          height,   // The crate's height
          volume,   // The volume of the crate
          cost,     // The cost to build the crate
          charge,   // The customer charge for the crate
          profit;   // The profit made on the crate

   // Set the desired output formatting for numbers.
   cout << setprecision(2) << fixed << showpoint;

   // Prompt the user for the crate's length, width, and height
   cout << "Enter the dimensions of the crate (in feet):\n";
   cout << "Length: ";
   cin >> length;
   cout << "Width: ";
   cin >> width;
   cout << "Height: ";
   cin >> height;

   // Calculate the crate's volume, the cost to produce it,
   // the charge to the customer, and the profit.
   volume = length * width * height;
   cost = volume * COST_PER_CUBIC_FOOT;
   charge = volume * CHARGE_PER_CUBIC_FOOT;
   profit = charge - cost;

   // Display the calculated data.
   cout << "The volume of the crate is ";
   cout << volume << " cubic feet.\n";
   cout << "Cost to build: $" << cost << endl;
   cout << "Charge to customer: $" << charge << endl;
   cout << "Profit: $" << profit << endl;
```

```
    cout << "Cost to build: $" << cost << endl;
    cout << "Charge to customer: $" << charge << endl;
    cout << "Profit: $" << profit << endl;
    return 0;
}
```