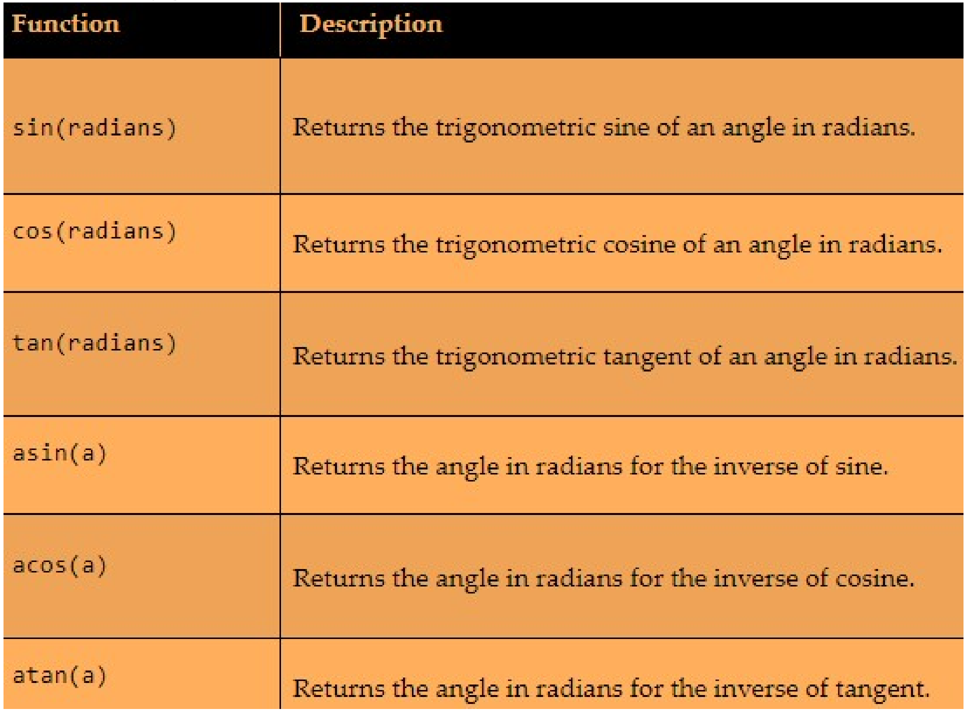
4.2 Math Functions

Monday, January 30, 2023 1:35 PM

- Already used pow(a,b) for a^b, and rand() function for random numbers
- This section will be trig, exponents, and service (min, max, and absolute)

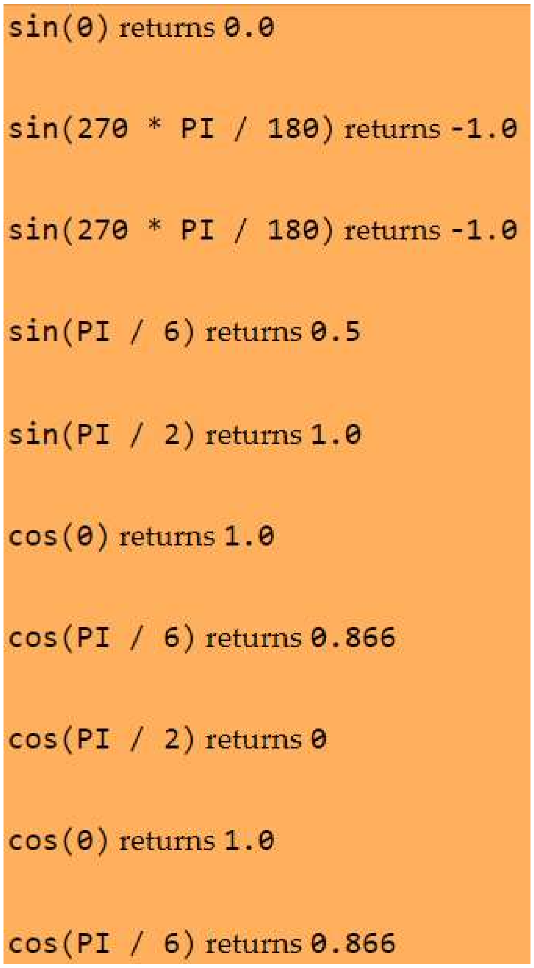
4.2.1 Trig Functions

Monday, January 30, 2023 4:57 PM

- cmath library got all of this
- 

Function	Description
`sin(radians)`	Returns the trigonometric sine of an angle in radians.
`cos(radians)`	Returns the trigonometric cosine of an angle in radians.
`tan(radians)`	Returns the trigonometric tangent of an angle in radians.
`asin(a)`	Returns the angle in radians for the inverse of sine.
`acos(a)`	Returns the angle in radians for the inverse of cosine.
`atan(a)`	Returns the angle in radians for the inverse of tangent.

- parameters for sin, cos, and tan is angle in radian
- Return val for asing & atan is angle in rad in range btwn –pi/2 and pi/2
- For acos, it's btwn 0 and pi
- 1 degree = pi/180 rad



`sin(0)` returns `0.0`

`sin(270 * PI / 180)` returns `-1.0`

`sin(270 * PI / 180)` returns `-1.0`

`sin(PI / 6)` returns `0.5`

`sin(PI / 2)` returns `1.0`

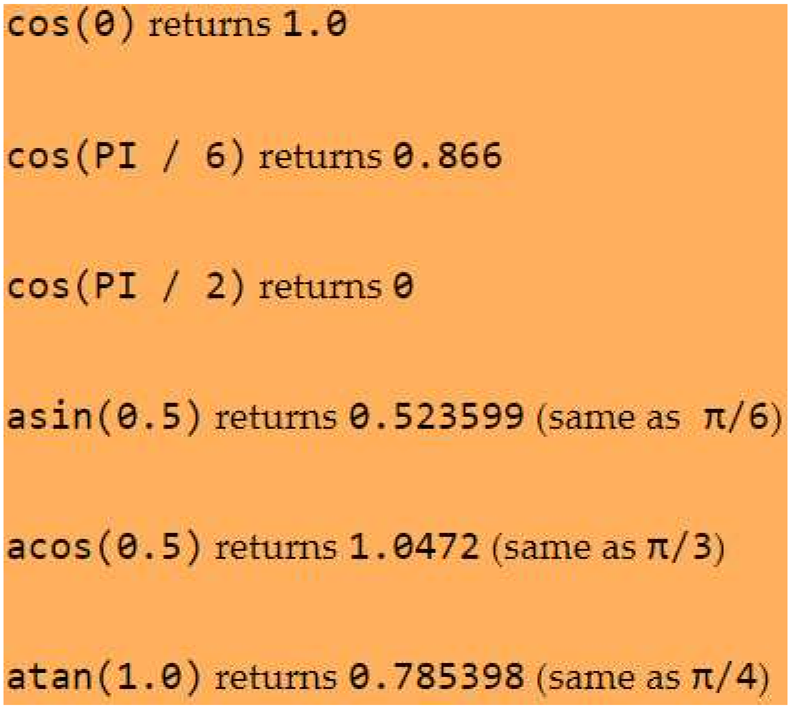
`cos(0)` returns `1.0`

`cos(PI / 6)` returns `0.866`

`cos(PI / 2)` returns `0`

`cos(0)` returns `1.0`

`cos(PI / 6)` returns `0.866`



`cos(0)` returns `1.0`

`cos(PI / 6)` returns `0.866`

`cos(PI / 2)` returns `0`

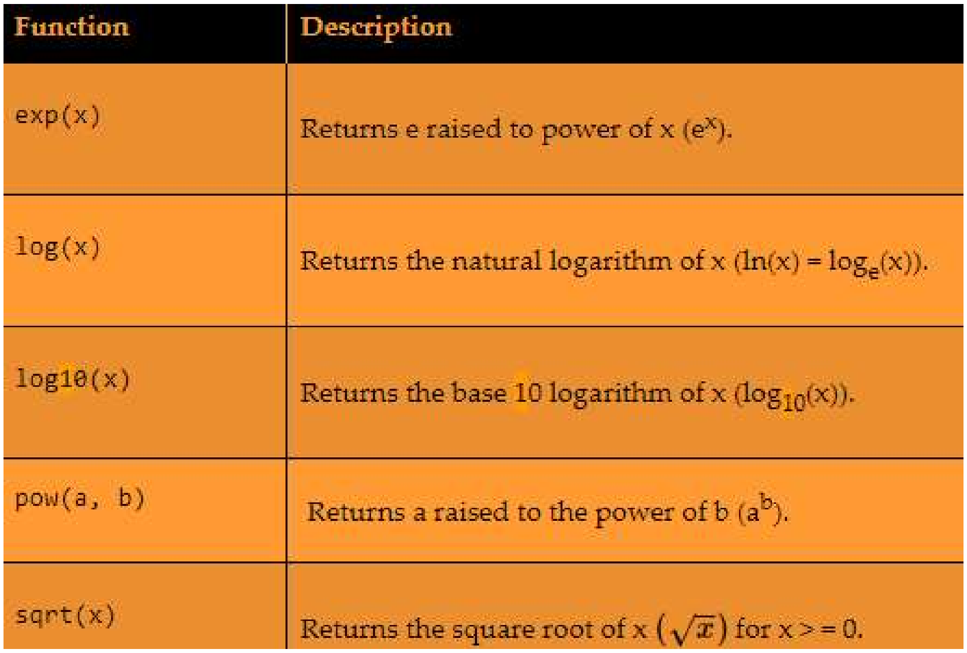
`asin(0.5)` returns `0.523599` (same as π/6)

`acos(0.5)` returns `1.0472` (same as π/3)

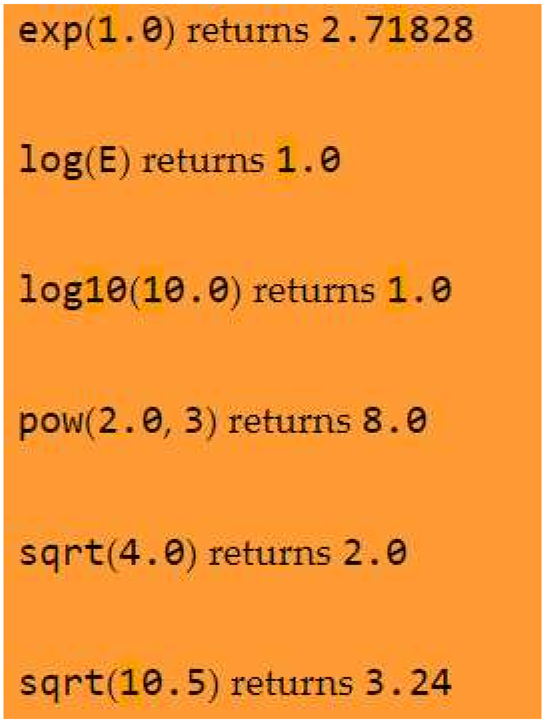
`atan(1.0)` returns `0.785398` (same as π/4)

4.2.2 Exponent Functions

Monday, January 30, 2023 5:09 PM

- Cmath lib has 5 functions
- 

Function	Description
`exp(x)`	Returns e raised to power of x (e^x).
`log(x)`	Returns the natural logarithm of x ($\ln(x) = \log_e(x)$).
`log10(x)`	Returns the base 10 logarithm of x ($\log_{10}(x)$).
`pow(a, b)`	Returns a raised to the power of b (a^b).
`sqrt(x)`	Returns the square root of x ($\sqrt{x}$) for x > = 0.

- Assume E is constant val 2.71828
- 

exp(1.0) returns 2.71828

log(E) returns 1.0

log10(10.0) returns 1.0

pow(2.0, 3) returns 8.0

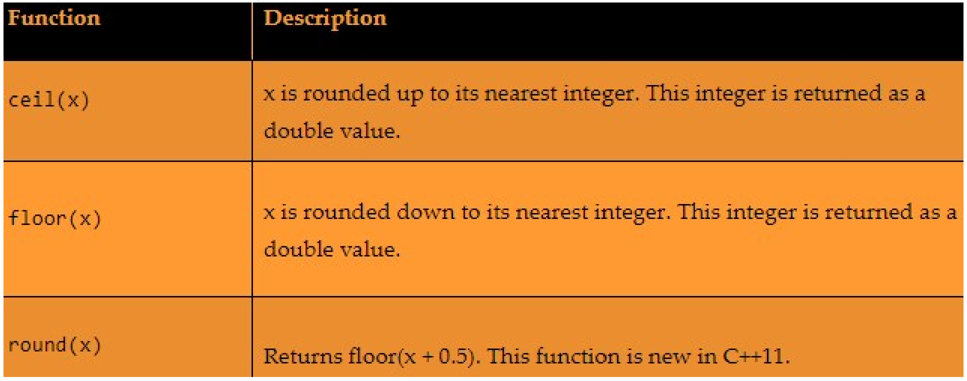
sqrt(4.0) returns 2.0

sqrt(10.5) returns 3.24

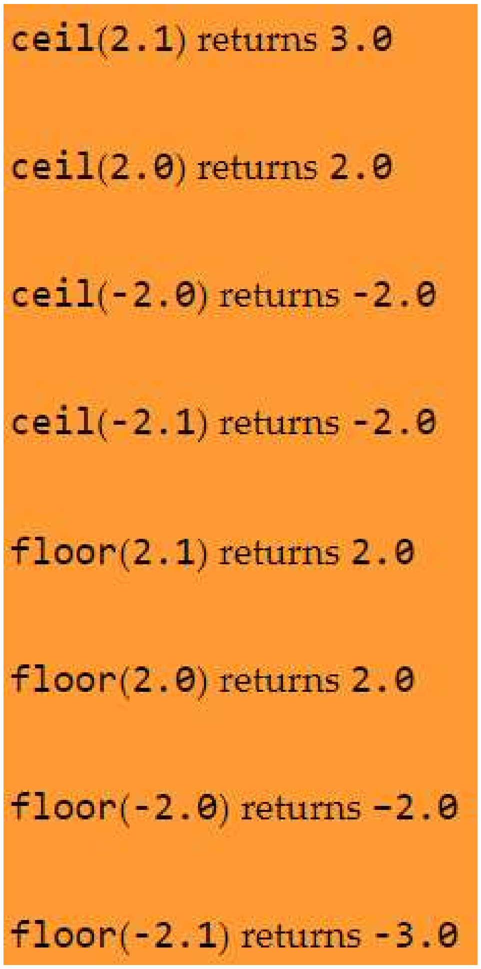
-

4.2.3 Rounding Functions

Monday, January 30, 2023 6:26 PM

- Cmath has 3 rounding functions
- 

Function	Description
`ceil(x)`	x is rounded up to its nearest integer. This integer is returned as a double value.
`floor(x)`	x is rounded down to its nearest integer. This integer is returned as a double value.
`round(x)`	Returns floor(x + 0.5). This function is new in C++11.

- 

`ceil(2.1)` returns `3.0`

`ceil(2.0)` returns `2.0`

`ceil(-2.0)` returns `-2.0`

`ceil(-2.1)` returns `-2.0`

`floor(2.1)` returns `2.0`

`floor(2.0)` returns `2.0`

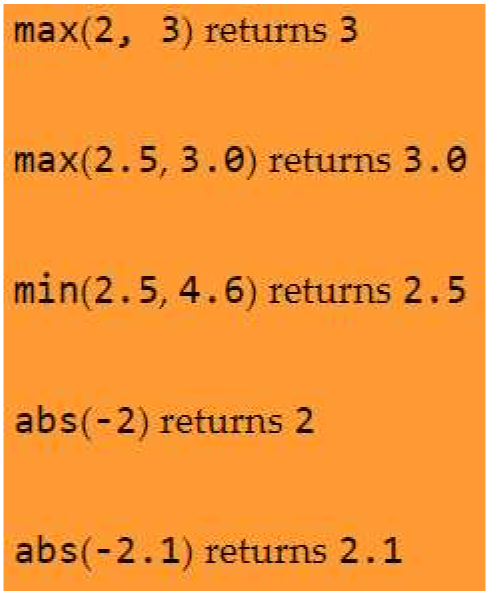
`floor(-2.0)` returns `-2.0`

`floor(-2.1)` returns `-3.0`

-
-

4.2.4 The min, max, and abs Functions

Monday, January 30, 2023 6:28 PM

- Min and max do min and max of 2 numbers (int, long, float, or double)
- 

  max(2, 3) returns 3

  max(2.5, 3.0) returns 3.0

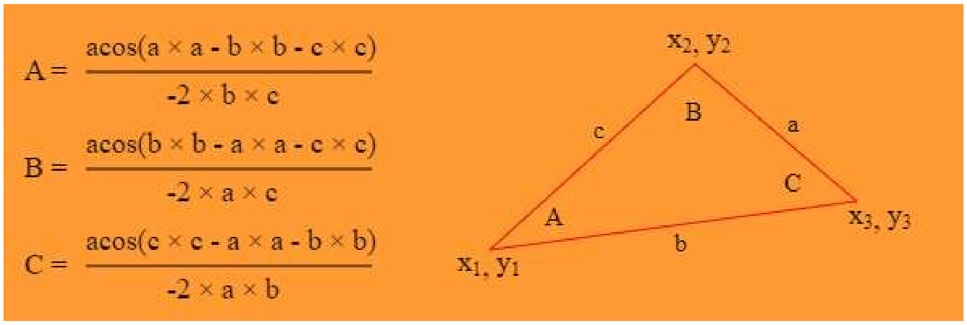
  min(2.5, 4.6) returns 2.5

  abs(-2) returns 2

  abs(-2.1) returns 2.1
-

4.2.5 Case Study: Computing Angles of a Triangle

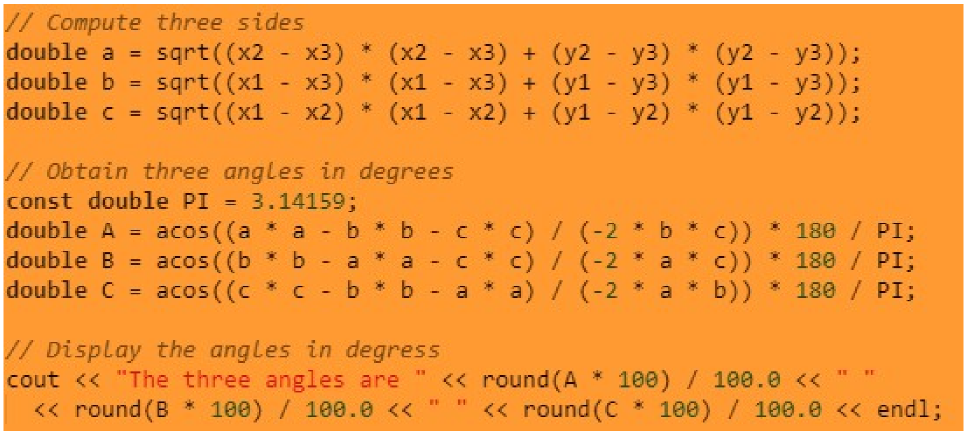
Monday, January 30, 2023 6:30 PM



$$A = \frac{\operatorname{acos}(a \times a - b \times b - c \times c)}{-2 \times b \times c}$$

$$B = \frac{\operatorname{acos}(b \times b - a \times a - c \times c)}{-2 \times a \times c}$$

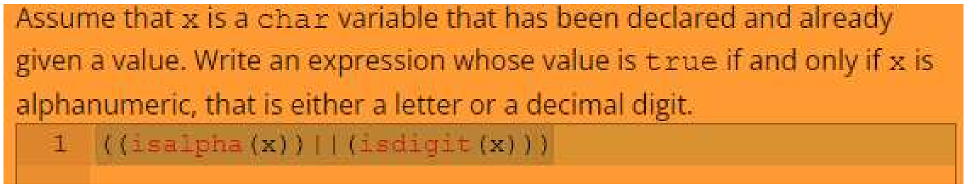
$$C = \frac{\operatorname{acos}(c \times c - a \times a - b \times b)}{-2 \times a \times b}$$



```
// Compute three sides
double a = sqrt((x2 - x3) * (x2 - x3) + (y2 - y3) * (y2 - y3));
double b = sqrt((x1 - x3) * (x1 - x3) + (y1 - y3) * (y1 - y3));
double c = sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));

// Obtain three angles in degrees
const double PI = 3.14159;
double A = acos((a * a - b * b - c * c) / (-2 * b * c)) * 180 / PI;
double B = acos((b * b - a * a - c * c) / (-2 * a * c)) * 180 / PI;
double C = acos((c * c - b * b - a * a) / (-2 * a * b)) * 180 / PI;

// Display the angles in degress
cout << "The three angles are " << round(A * 100) / 100.0 << " "
  << round(B * 100) / 100.0 << " " << round(C * 100) / 100.0 << endl;
```

4.3 Character Data Type and Operations

Monday, January 30, 2023 6:37 PM

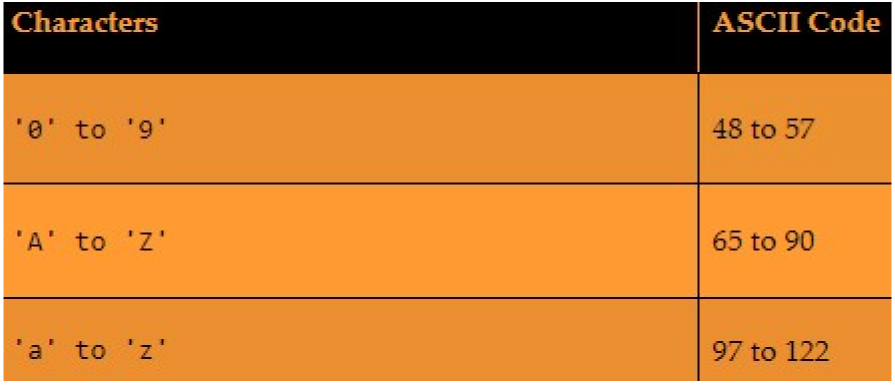
- Characters in C++ as char, rep single character, character literal enclosed in 'A' (single quotation marks)
- 

  Assume that x is a char variable that has been declared and already given a value. Write an expression whose value is true if and only if x is alphanumeric, that is either a letter or a decimal digit.

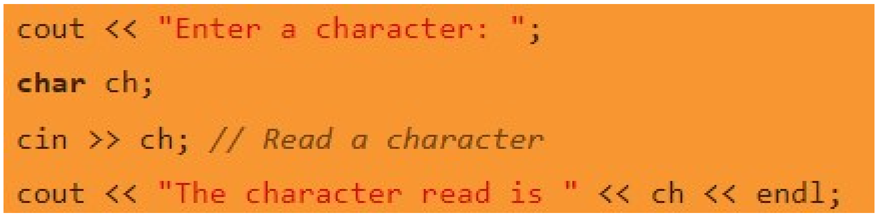
  ```
  ((isalpha(x))||(isdigit(x)))
  ```

4.3.1 & 2 ASCII Code & Reading char from Keyboard

Monday, January 30, 2023 6:39 PM

- Binary in computer, mapping char to its binary rep called encoding, how characters encoded defined by encoding scheme
- Most pc's use ASCII, 8-bit encoding to rep all uppercase & lowercase letters, digits, punctuation marks, and ctrl characters
- Most sys, char type is 1 byte
- 

Characters	ASCII Code
`'0' to '9'`	48 to 57
`'A' to 'Z'`	65 to 90
`'a' to 'z'`	97 to 122

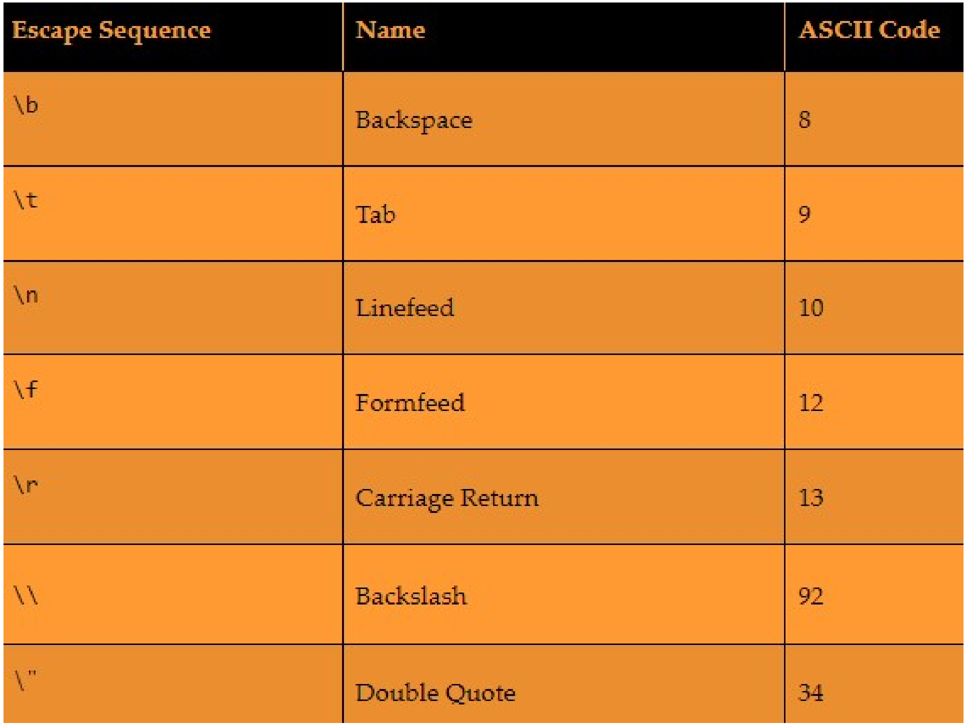
-
- 

```
cout << "Enter a character: ";
char ch;
cin >> ch; // Read a character
cout << "The character read is " << ch << endl;
```

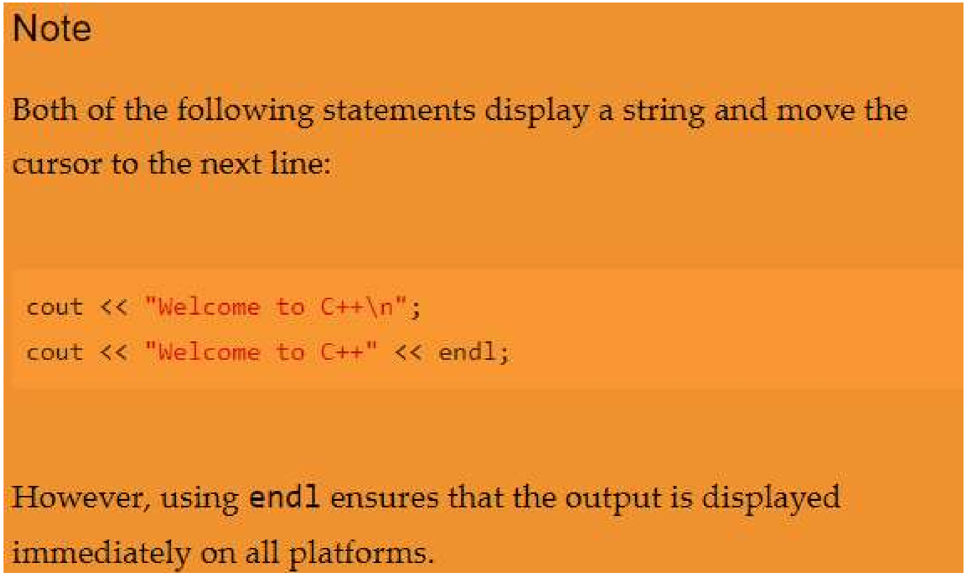
-

4.3.3 Escape Sequence for Special Characters

Monday, January 30, 2023 6:43 PM

- Can't print quotations by just putting them in other quotations
- That's why we have escape sequences, has backslash (\) w/ character/combo of digits after
- 

Escape Sequence	Name	ASCII Code
\b	Backspace	8
\t	Tab	9
\n	Linefeed	10
\f	Formfeed	12
\r	Carriage Return	13
\\	Backslash	92
\"	Double Quote	34

- Characters ' ' , '\t', '\f', '\r' ,and '\n' known as whitespace characters
- 

Note

Both of the following statements display a string and move the cursor to the next line:

```
cout << "Welcome to C++\n";
cout << "Welcome to C++" << endl;
```

However, using `endl` ensures that the output is displayed immediately on all platforms.

-

4.3.4 Casting between char and Numeric Types

Monday, January 30, 2023 6:49 PM

- Char can be cast into any numeric type (and vice versa)
- When int cast to char, only bottom 8 bits used, other pt is ignored
- When float cast to char, floating-pt val is 1st cast to int, then to char (65.25 -> 65 -> c)
- When char to numeric type, char's ASCII cast to specified numeric type ('A' -> 65)
- Char type treated as integer of byte size, all numeric operators can be applied to char operands cus auto cast to numb
- static_cast<char>(value) explicitly casts a numeric value into a character
-

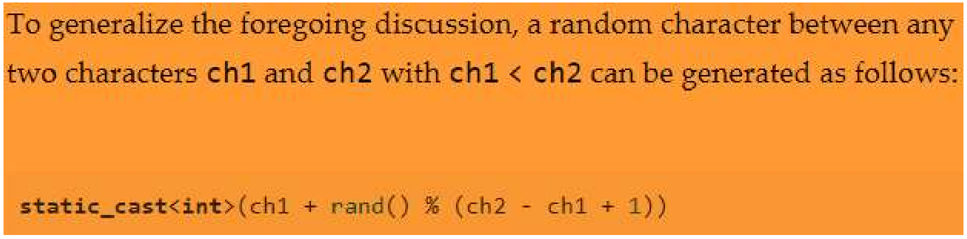
4.3.5 Comparing and Testing Characters

Monday, January 30, 2023 7:01 PM

- 2 chars can be compared using the relational operators like comparing 2 numbs

4.4 Case Study: Generating Random Characters

Monday, January 30, 2023 7:18 PM

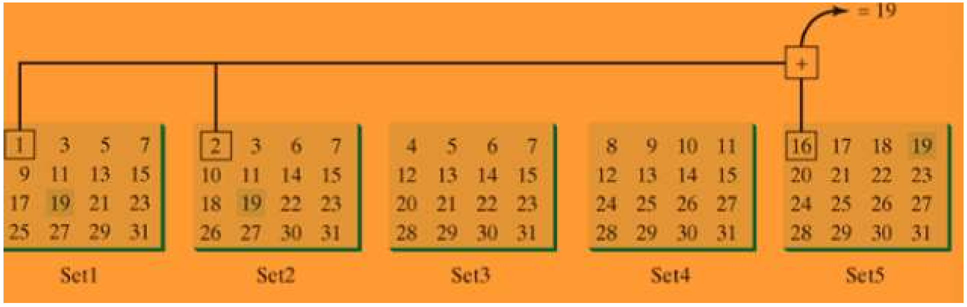
- ASCII has 0-127 chars
- rand() & srand(seed) for a seed
- In general
  - a + rand() % b
  - Will return a random integer btwn a and a+b-1
  - 50 + rand() % 50
  - Returns a random integer btwn 50 and 99
- So random int btwn 0 and 127 is:
  - rand() % 128
- Random lowercase is :
  - static_cast<int>('a') + rand() % (static_cast<int>('z') - static_cast<int> ('a') +1)
  - Also can be simplified to :
  - 'a' + rand() % ('z' - 'a' + 1)
- 

  To generalize the foregoing discussion, a random character between any two characters **ch1** and **ch2** with **ch1** < **ch2** can be generated as follows:

  ```
  static_cast<int>(ch1 + rand() % (ch2 - ch1 + 1))
  ```
- Header for rand is
  - #include <cstdlib>

4.5 Case Study: Guessing Birthdays

Monday, January 30, 2023 7:27 PM

- Ask 5 questions, get day of birthday (like day of the month, not the month or year)
- Ask which sets their bday appears on:
- 

1	3	5	7
9	11	13	15
17	19	21	23
25	27	29	31

Set1

2	3	6	7
10	11	14	15
18	19	22	23
26	27	30	31

Set2

4	5	6	7
12	13	14	15
20	21	22	23
28	29	30	31

Set3

8	9	10	11
12	13	14	15
24	25	26	27
28	29	30	31

Set4

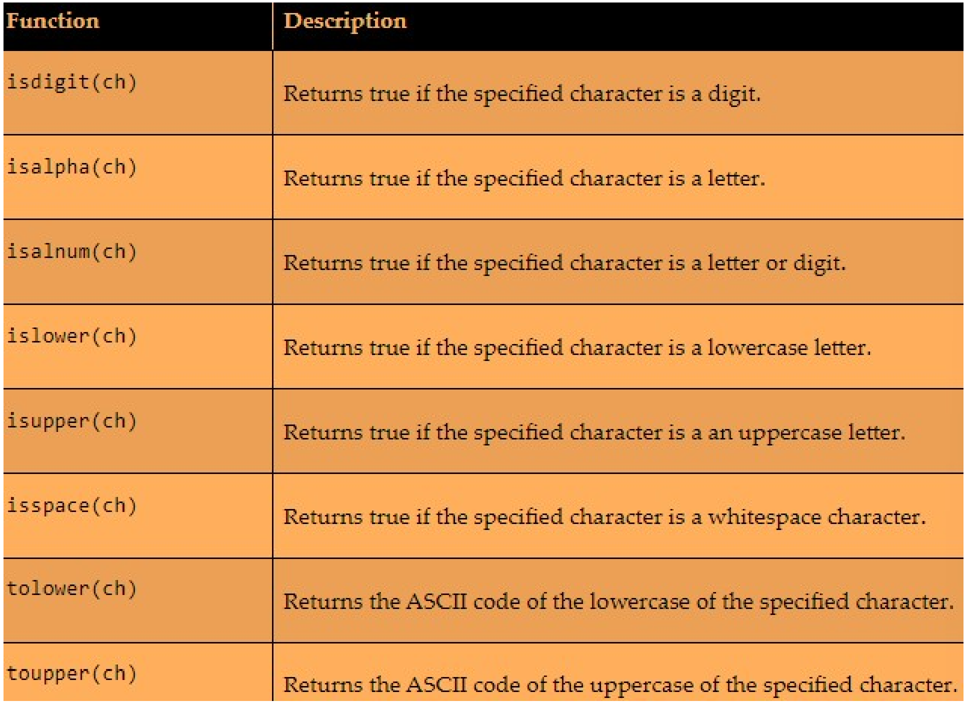
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31

Set5

-

4.6 Character Functions

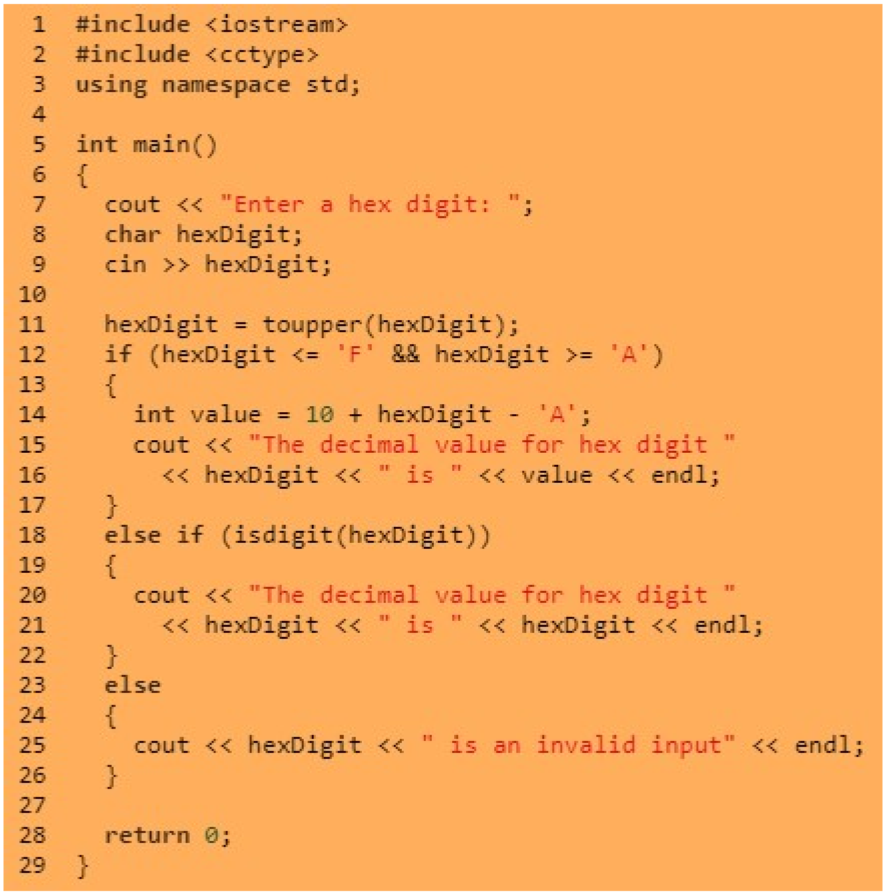
Monday, January 30, 2023 7:35 PM

- Many functions for testing a char & converting char, <cctype>
- Testing functions test single char & return T/F (actually return int, nonzero = true, 0 = false)
- 

Function	Description
`isdigit(ch)`	Returns true if the specified character is a digit.
`isalpha(ch)`	Returns true if the specified character is a letter.
`isalnum(ch)`	Returns true if the specified character is a letter or digit.
`islower(ch)`	Returns true if the specified character is a lowercase letter.
`isupper(ch)`	Returns true if the specified character is a an uppercase letter.
`isspace(ch)`	Returns true if the specified character is a whitespace character.
`tolower(ch)`	Returns the ASCII code of the lowercase of the specified character.
`toupper(ch)`	Returns the ASCII code of the uppercase of the specified character.

4.7 Case Study: Converting a Hexadecimal Digit to a Decimal Value

Tuesday, January 31, 2023 1:50 PM

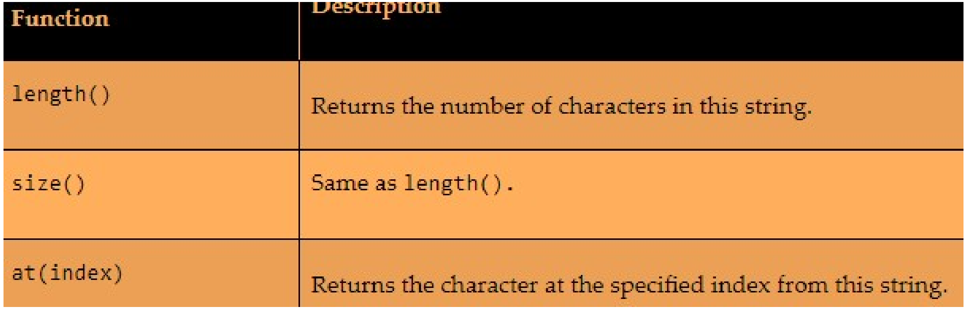
- Hex numb sys has 16 digits: 0-9, A-F
  - A=10, B=11, …, F=15
-
- 

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

int main()
{
  cout << "Enter a hex digit: ";
  char hexDigit;
  cin >> hexDigit;

  hexDigit = toupper(hexDigit);
  if (hexDigit <= 'F' && hexDigit >= 'A')
  {
    int value = 10 + hexDigit - 'A';
    cout << "The decimal value for hex digit "
      << hexDigit << " is " << value << endl;
  }
  else if (isdigit(hexDigit))
  {
    cout << "The decimal value for hex digit "
      << hexDigit << " is " << hexDigit << endl;
  }
  else
  {
    cout << hexDigit << " is an invalid input" << endl;
  }

  return 0;
}
```

-

4.8 The string Type

Tuesday, January 31, 2023 1:58 PM

- Char type reps 1 character, to rep string of char, use string
- string message = "Programming is fun";
- string type not primitive, its object type, so it reps an object
- Objects defined using classes, so need <string> header file
- Object also known as instance of class
- 

Function	Description
`length()`	Returns the number of characters in this string.
`size()`	Same as `length()`.
`at(index)`	Returns the character at the specified index from this string.

- Functions in string class only invoked from specific string instance, so called instance functions (aka object member functions)
-

4.8.1 String Index and Subscript Operator

Wednesday, February 1, 2023 10:08 PM

- s.at(index) function can get certain chars in string s
- index btwn 0 and s.length()-1
- Subscript operator to get char @ specified index in string, use syntax;
  - stringName[index]
- Can use this to get and mod the char in a string

4.8.2 Concatenating String

Wednesday, February 1, 2023 10:11 PM

- + for concatenating 2 strings
  - string s3 = s1 + s2;
- += can also be used, adds to the string really
  - message = "Welcome to C++"
  - message += "and programming is fun";
- Cannot concatenate 2 string literals in C++

4.8.3 Comparing Strings

Wednesday, February 1, 2023 10:17 PM

- Can use relational operators ==, !=, <, <=, >, >= to compare 2 strings
- Dun by comparing 1 character at a time, left to right
  - s1= "ABC" and s2 = "DEFG"
  - s1 < s2 is true

4.8.4 Reading Strings

Wednesday, February 1, 2023 10:20 PM

- Use cin, but input ends w/ a whitespace character (So input of New York will just input New)
- Use getline function in string header file, reads string from keyboard in this syntax:
    - getline(cin, s, delimitCharacter)
- Function stops reading when delimiter encountered, delimiter read but not stored
- If delimiterCharacter is \n, don't have to keep it there
    - getline(cin,city,'\n');
    - Is same as
    - getline(cin,city);
- Call input using cin as "token-based input" bc it reads elements separated by whitespace chars
- Input using the getline function as "line-based input" bc reads entire line
- Don't use l-b input after t-b input, usually bc t-b stops reading b4 a line break char, and the l-b stops after reading a \n, which is literally the next character

4.9 Case Study: Revising the Lottery Program Using Strings

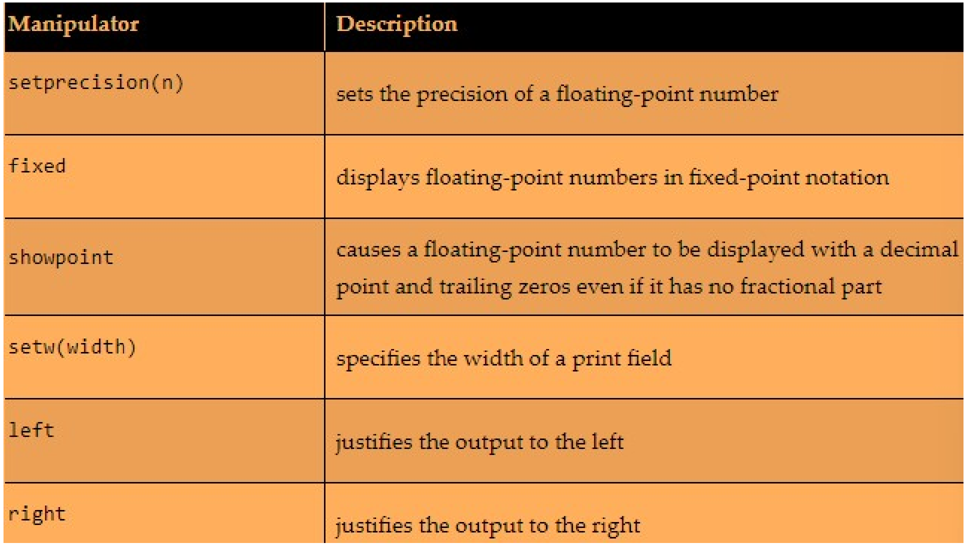
Thursday, February 2, 2023 9:30 AM

- The pgrm initially generates a 2-digit numb, finds if win to the rule:
    1. If user input lottery numb in exact order, award is $10,000
    2. If all digits user entered match digits of lottery numb, $3,000
    3. If 1 digit in user input matches digit in lottery numb, $1,000
- Now we'll make pgrm generate random 2-digit string, and gets user input as string (not number)
- The pgrm now makes random digit, casts to char, concatenates the char to the string lottery, then makes second random digit and cast it to char and concatenates char to string
- Checks guesses in order:
    1. Check whether guess matches lottery exactly
    2. Check whether reversal of guess matches lottery
    3. Check if 1 digit is in lottery
    4. Else, displays: "Sorry, no match"

4.10 Formatting Console Output

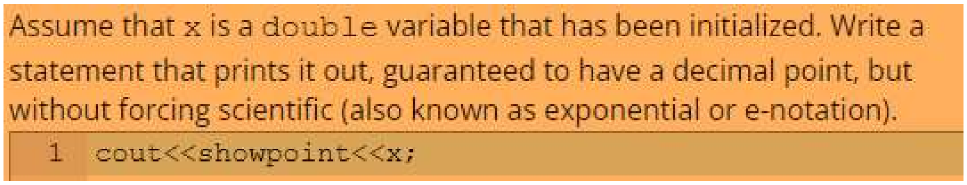
Thursday, February 2, 2023 9:41 AM

- Might wanna display stuff in certain way
- Can format using the iomanip header file



Manipulator	Description
`setprecision(n)`	sets the precision of a floating-point number
`fixed`	displays floating-point numbers in fixed-point notation
`showpoint`	causes a floating-point number to be displayed with a decimal point and trailing zeros even if it has no fractional part
`setw(width)`	specifies the width of a print field
`left`	justifies the output to the left
`right`	justifies the output to the right

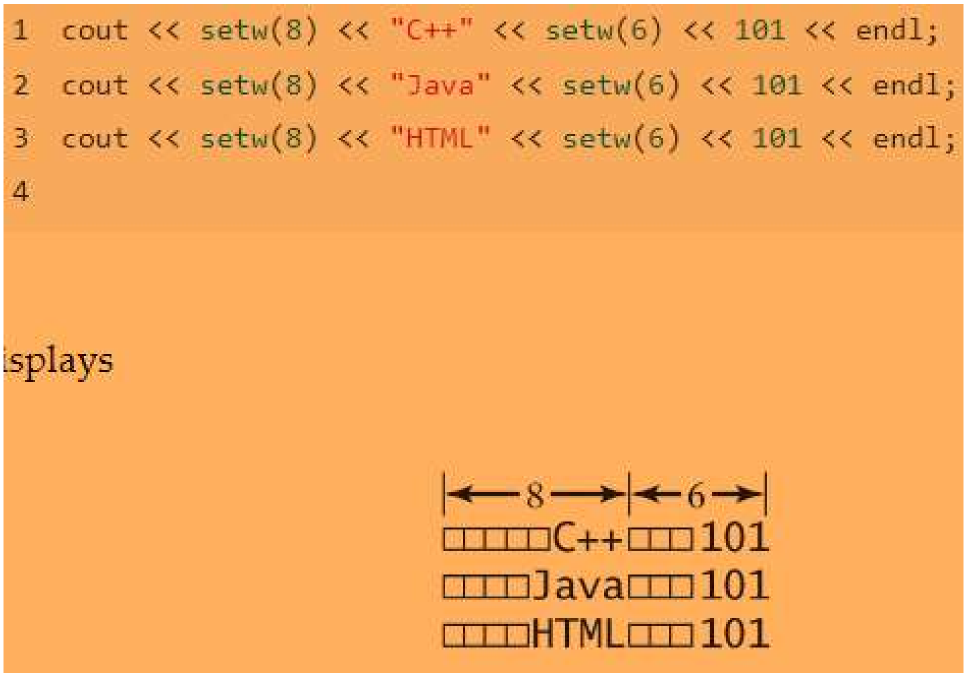
- setprecision(n) manipuplator lets specify tot numb of digits displayed for floating-point numb, n is number of significant digits (tot # digits appear b4 & after decimal point), if numb to be displayed has more digits, it rounded
    - Double number = 12.34567;
    - cout<<setprecision(5)<<number<<endl;
    - Will print out:
    - 12.346
    - Also, the setprecision will stay in effect until precision is changed
    - If width isn't good enough for integer, the setprecision is ignore:
    - cout<<setprecision(3)<<23456<<endl;
    - Will print out:
    - 23456
- Sometimes comp auto display large # in scientific notation, but can use fixed manipulator to force # to display nonscientific notation w/ fixed # of digits after the decimal point
- By default, fixed # of digits after decimal pt is 6, can change it using fixed and setprecision manipulators:
    - Cout<<fixed<<setprecision(4)<<…
- showpoint manipulator shows all points, so used with setprecision usually:
    - cout<<setprecision(6)<<showpoint<<1.23<<endl;
    - Will print out:
    - 1.23000



Assume that `x` is a `double` variable that has been initialized. Write a statement that prints it out, guaranteed to have a decimal point, but without forcing scientific (also known as exponential or e-notation).

```
cout<<showpoint<<x;
```

- setw(width) manipulator specifies min number of positions needed for an output

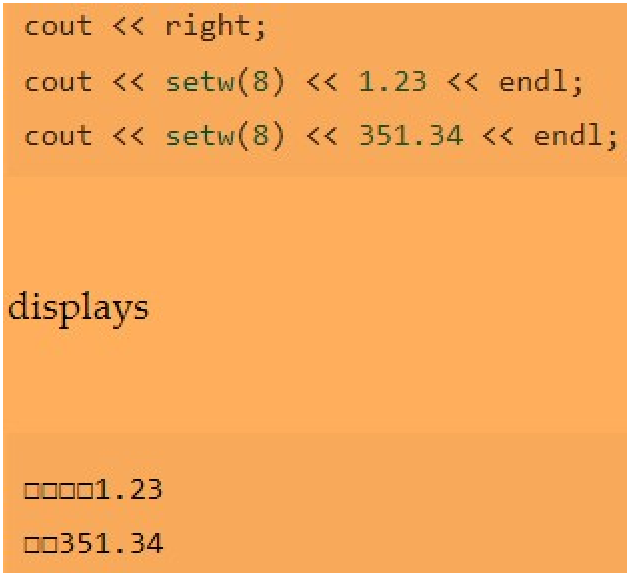


```
cout << setw(8) << "C++" << setw(6) << 101 << endl;
cout << setw(8) << "Java" << setw(6) << 101 << endl;
cout << setw(8) << "HTML" << setw(6) << 101 << endl;
```

isplays

```
□□□□□C++□□□101
□□□□Java□□□101
□□□□HTML□□□101
```

- Output is right justified in specified spaces, also, setw only affects next output, so:
  - Cout<<setw(8)<<"C++"<<101<<endl;
  - Will print out:
  - _____C++101
- If item need more space than specified width, width auto increased
- Left and right manipulators to justify the output, usually use with setw()



```
cout << right;
cout << setw(8) << 1.23 << endl;
cout << setw(8) << 351.34 << endl;
```

displays

```
□□□□1.23
□□351.34
```

- Set once, used for all following

4.11 Simple File Input & Output

Thursday, February 2, 2023 10:24 AM

- Cin to read input, cout write output to console, also read/write data from/to file
- Writing to a file:
    1. First, declare object of the ofstream type:
        i. ofstream output;
    2. To specify file, invoke open function from output:
        i. output.open("number.txt");
    3. This makes a file named numbers.txt, if this file already exists, then contents are destroyed and new file created, invoking open function is to associate file w/ the stream
    4. Can also create file output object and open the file in 1 statement:
        i. ofstream output("numbers.txt");
    5. To write data, use the stream insertion operator (<<) just like cout
        i. output<<95<<" "<<56<<" "<<34<<endl;
    6. After dun, invoke close function from output:
        i. Output:close();
    7. Needs to invoke close so data written to file b4 pgrm exits
- To read data from file:
    1. Declare an object of the ifstream type:
        i. ifstream input;
    2. Then specify file, invoke open function from input:
        i. input.open("numbers.txt");
    3. This statement opens file names numbers.txt for input, if it doesn't exist, errors
    4. Can also create a file input object and open the file in 1 statement:
        i. ifstream input("numbers.txt");
    5. To read data, use stream extraction operator (>>), same as cin object
        i. input>>score1;
    6. After dun, invoke close function from input:
        i. input.close();
- ifstream needs fstream header file

Ch 4 Summary

Thursday, February 2, 2023 10:55 AM

1. C++ provides the mathematical functions sin, cos, tan, asin, acos, atan, exp, log, log10, pow, sqrt, cell, floor, min, max, and abs for performing mathematical functions.
2. Character type (char) represents a single character.
3. The character \ is called an escape character and an escape sequence starts with the escape character followed by another character or a combination of digits.
4. C++ allows you to use escape sequences to represent special characters such as '\t' and '\n'.
5. The characters ' ', '\t', '\f', '\r', and '\n' are known as the whitespace characters.
6. C++ provides the functions isdigit, isalpha, isalnum, islower, isupper, isspace for testing whether a character is a digit, letter, digit or letter, lowercase, uppercase, and whitespace. It also contains the tolower and toupper functions for returning a lowercase or uppercase letter.
7. A string is a sequence of characters. A string value is enclosed in matching double quotes ("). A character value is enclosed in matching single quotes (').
8. You can declare a string object using the string type. A function that is invoked from a specific object is called an instance function.
9. You can get the length of a string by invoking its length() function, and retrieve a character at the specified index in the string using the at(index)
10. You can use the subscript operator to retrieve or modify the character in a string and can use the + operator to concatenate two strings,
11. You can use the relational operators to compare two strings.
12. You can format output using stream manipulators defined in the iomanip header.
13. You can create an ifstream object for reading data from a file and an ofstream object for writing data to a file.