CIS 22A – Lecture 6

Manish Goel

Unsigned (UINT) and Signed (int) Integers

4-bit Unsigned Integers						4-bit Signed Integers				
0	0000		8	1000		-8	1000		0	0000
1	0001		9	1001		-7	1001		1	0001
2	0010		10	1010		-6	1010		2	0010
3	0011		11	1011		-5	1011		3	0011
4	0100		12	1100		-4	1100		4	0100
5	0101		13	1101		-3	1101		5	0101
6	0110		14	1110		-2	1110		6	0110
7	0111		15	1111		-1	1111		7	0111

Multiple Assignments

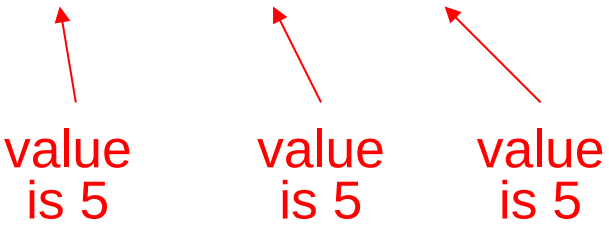
- The = can be used to assign a value to multiple variables:

```
x = y = z = 5;
```

- Value of = is the value that is assigned

- Associates right to left:

```
x = (y = (z = 5));
```



Combined Assignment

- `sum = sum + 1;` // adds 1 to the variable **`sum`**
- `res = res * 2;` // multiplies variable **`res`** by 2

- Combined assignment operators provide shorthand for such operations.
- `sum = sum + 1;` ➔ `sum += 1;`
- `Res = res * 2;` ➔ `res *= 2;`

- For addition and subtraction, shorthand for incrementing / decrementing
- `x = x + 1` ➔ `x += 1` ➔ `x++`
- `y = y - 1` ➔ `x -= 1` ➔ `y--`

Table 3-9

Operator	Example Usage	Equivalent to
+=	`x += 5;`	`x = x + 5;`
-=	`y -= 2;`	`y = y - 2;`
*=	`z *= 10;`	`z = z * 10;`
/=	`a /= b;`	`a = a / b;`
%=	`c %= 3;`	`c = c % 3;`

Formatting Output

- Requires the `iomanip` library
- Control output display for numeric and string data
  - Size (width), Position, # of digits, Alignment

- `setw(x)` : print a field of at least x spaces
- `fixed` : use decimal notation
- `setprecision(x)` : print x significant digits after decimal
- `fixed & setprecision(x)` : print x digits after decimal
- `showpoint` : always print decimal point with trailing zeroes
- `left` : print values to be left justified (aligned)
- `right` : print values to be right justified (aligned)

`string` and character inputs

- `cin` with `>>` ignores whitespace characters (*spaces, tabs, line breaks*)

```
string inputStr;
cin >> inputStr;        //Enter "This is my input"
```

Only "`This`" stored in `inputStr`

- To accept whitespace characters, use `getline(cin, x)`

```
string inputStr;
getline(cin, inputStr);   //Enter as befoer
```

"`This is my input`" stored in `inputStr`

`string` and character inputs - 2

- Use cin.get(ch) to accept a single space character

  ```
  cin.get():
  cin.get(ch);
  ```

  Will read the next character entered, even whitespace

- Mixing `cin >>` and `cin.get()` in the same program can cause input errors that are hard to detect

- To skip over unneeded characters that are still in the keyboard buffer, use `cin.ignore()`:

```
cin.ignore(); // skip next char
cin.ignore(10, '\n'); // skip the next
                // 10 char. or until a '\n'
```

`string` Member Functions and Operators

- To find the length of a string:

```
string state = "California"
int size = state.length();
```

- `sizeof(string)` = 4 bytes on 32-bit and 64-bit machines → size of the address where the string is stored in memory.

- To concatenate (join) multiple strings:

```
resultStr = inputStr1 + inputStr 2;
```

if `inputStr1 = "Hello"`
and `inputStr2 = "World"`
then `resultStr = "HelloWorld"`

Also, `resultStr += "!!!";`
means `resultStr = "HelloWorld!!!"`

More Mathematical Library Functions

- Require `cmath` header file
- Take `double` as input, return a `double`
- Commonly used functions:

`sin`	Sine
`cos`	Cosine
`tan`	Tangent
`pow`	Exponent
`sqrt`	Square root
`log`	Natural (e) log
`log10`	Log Base (10)
`abs`	Absolute value (takes and returns an int)

More Mathematical Library Functions

- These require `cstdlib` header file
- `rand()`: returns a random number (`int`) between `0` and the largest int the compute holds. Yields same sequence of numbers each time program is run
- `srand(x)`: initializes random number generator with `unsigned int x`