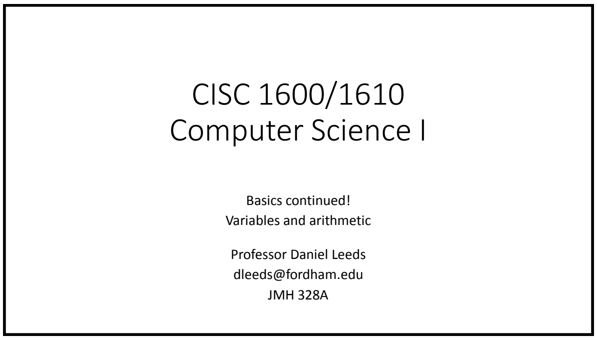
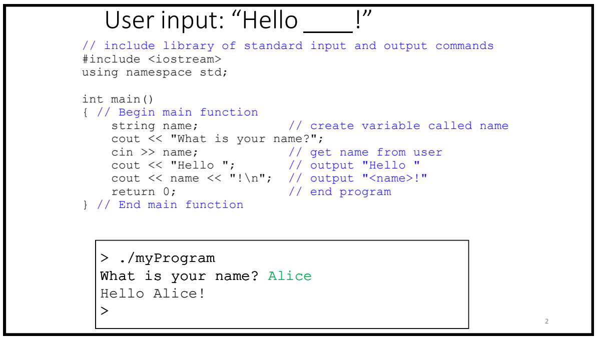


CISC 1600/1610
Computer Science I

Basics continued!
Variables and arithmetic

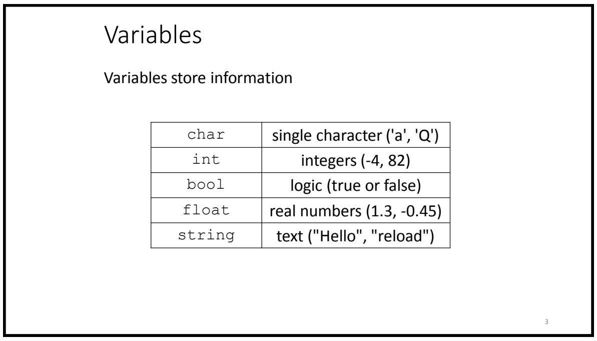
Professor Daniel Leeds
dleeds@fordham.edu
JMH 328A



User input: “Hello ____!”

```
// include library of standard input and output commands
#include <iostream>
using namespace std;

int main()
{ // Begin main function
    string name;            // create variable called name
    cout << "What is your name?";
    cin >> name;            // get name from user
    cout << "Hello ";       // output "Hello "
    cout << name << "!\n";  // output "<name>!"
    return 0;               // end program
} // End main function
```

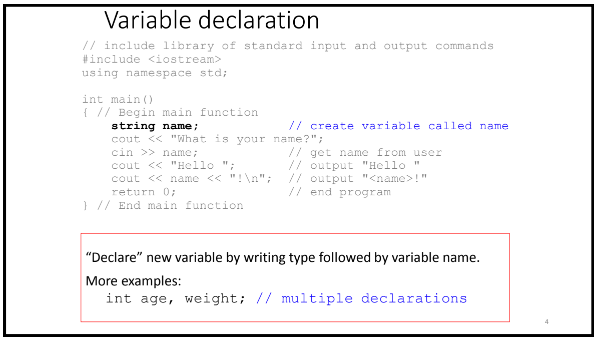
```
> ./myProgram
What is your name? Alice
Hello Alice!
>
```



Variables

Variables store information

char	single character ('a', 'Q')
int	integers (-4, 82)
bool	logic (true or false)
float	real numbers (1.3, -0.45)
string	text ("Hello", "reload")



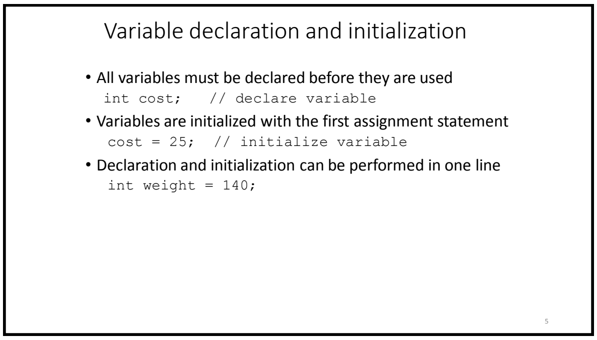
Variable declaration

```
// include library of standard input and output commands
#include <iostream>
using namespace std;

int main()
{ // Begin main function
    string name;            // create variable called name
    cout << "What is your name?";
    cin >> name;            // get name from user
    cout << "Hello ";       // output "Hello "
    cout << name << "!\n";  // output "<name>!"
    return 0;               // end program
} // End main function
```

“Declare” new variable by writing type followed by variable name.

More examples:

```
  int age, weight; // multiple declarations
```



Variable declaration and initialization

- All variables must be declared before they are used

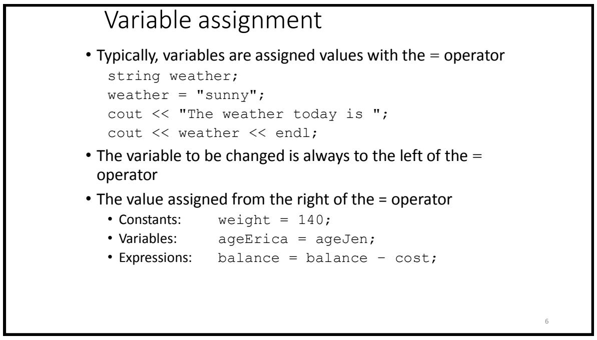
```
int cost;   // declare variable
```

- Variables are initialized with the first assignment statement

```
cost = 25;   // initialize variable
```

- Declaration and initialization can be performed in one line

```
int weight = 140;
```

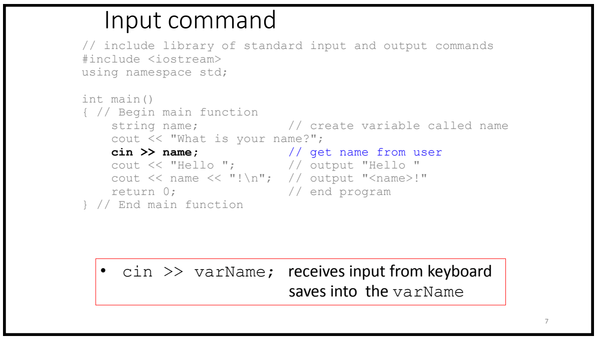


Variable assignment

- Typically, variables are assigned values with the = operator

```
string weather;
weather = "sunny";
cout << "The weather today is ";
cout << weather << endl;
```

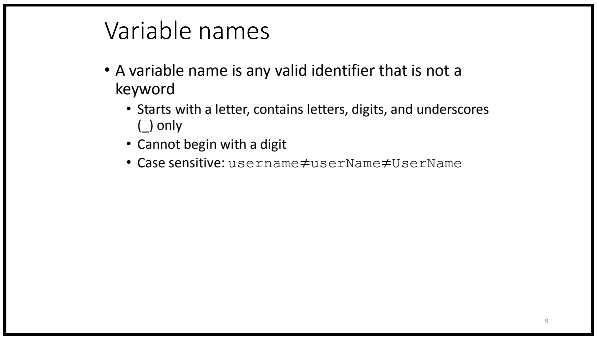
- The variable to be changed is always to the left of the = operator
- The value assigned from the right of the = operator
  - Constants: `weight = 140;`
  - Variables: `ageErica = ageJen;`
  - Expressions: `balance = balance - cost;`



Input command

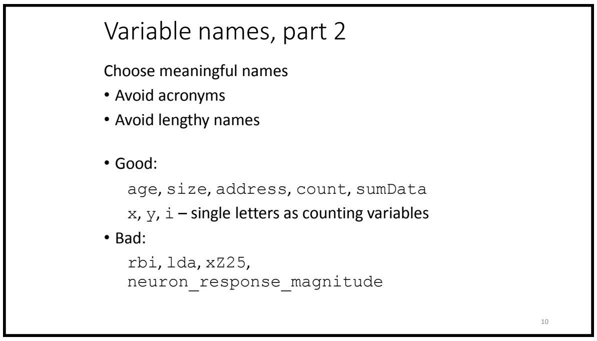
```
// include library of standard input and output commands
#include <iostream>
using namespace std;

int main()
{ // Begin main function
    string name;            // create variable called name
    cout << "What is your name?";
    cin >> name;            // get name from user
    cout << "Hello ";       // output "Hello "
    cout << name << "!\n";  // output "<name>!"
    return 0;               // end program
} // End main function
```

- `cin >> varName;` receives input from keyboard saves into the `varName`



Variable names

- A variable name is any valid identifier that is not a keyword
  - Starts with a letter, contains letters, digits, and underscores (_) only
  - Cannot begin with a digit
  - Case sensitive: `username≠userName≠UserName`



Variable names, part 2

Choose meaningful names

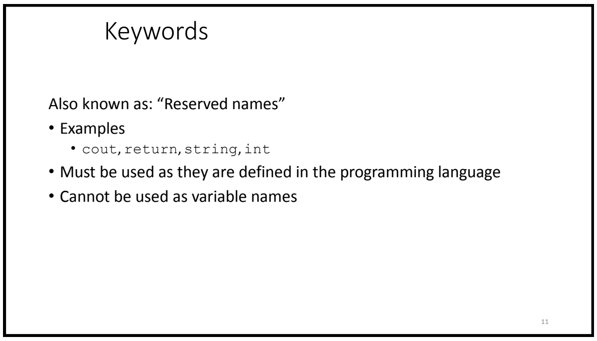
- Avoid acronyms
- Avoid lengthy names

- Good:

  `age`, `size`, `address`, `count`, `sumData`

  `x`, `y`, `i` – single letters as counting variables
- Bad:

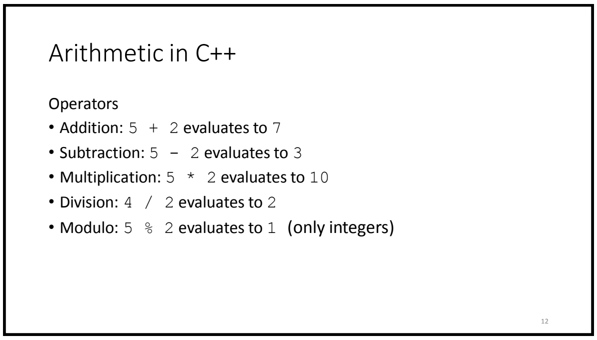
  `rbi`, `lda`, `xZ25`,
  `neuron_response_magnitude`



Keywords

Also known as: “Reserved names”

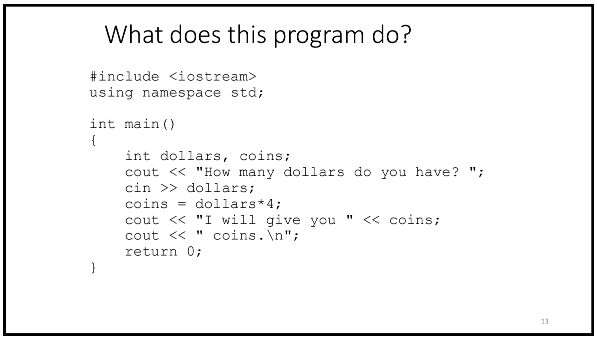
- Examples
  - `cout`, `return`, `string`, `int`
- Must be used as they are defined in the programming language
- Cannot be used as variable names



Arithmetic in C++

Operators

- Addition: `5 + 2` evaluates to `7`
- Subtraction: `5 - 2` evaluates to `3`
- Multiplication: `5 * 2` evaluates to `10`
- Division: `4 / 2` evaluates to `2`
- Modulo: `5 % 2` evaluates to `1` (only integers)



What does this program do?

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

int main()
{
    int dollars, coins;
    cout << "How many dollars do you have? ";
    cin >> dollars;
    coins = dollars*4;
    cout << "I will give you " << coins;
    cout << " coins.\n";
    return 0;
}
```

Order of operations

- First: Parentheses
- Second: Multiplication, Division, Modulo
- Third: Add, Subtract

- Evaluate from Left to Right
- Evaluate inner-most parentheses before outer ones

```
int a = ( 4 * ( 5 + 2 ) - 4 ) / 4;
```

Assignment operators

```
int a = 6;
```

- Standard assignment: `a = 3;`
- Other assignments:
  - `a += 3;  // a = a + 3;`
  - `a -= 3;  // a = a - 3;`
  - `a *= 3;  // a = a * 3;`
  - `a /= 3;  // a = a / 3;`
  - `a %= 3;  // a = a % 3;`

Increment and decrement

```
int c = 12;
```

- Increment by 1: `c++` evaluates to `c + 1`

- Decrement by 1: `c--` evaluates to `c - 1`

The binary representation

- `int age = 65;` assigns a binary code to memory: 00000000000000000000000001000001
- `char grade = 'A';` assigns a binary code to memory: 01000001
- Every variable value is a number in binary, C++ interprets the binary number based on the variable type

Interpreting binary

Base 10

253 -> 253

2x100+5x10+3x1

Base 2

128 64 32 16 8 4 2 1

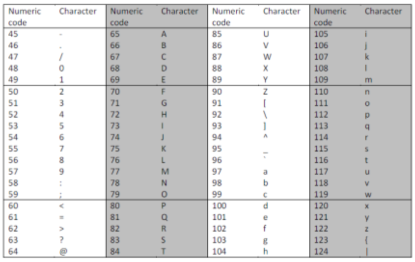
- - - - - - - -

00001001=?

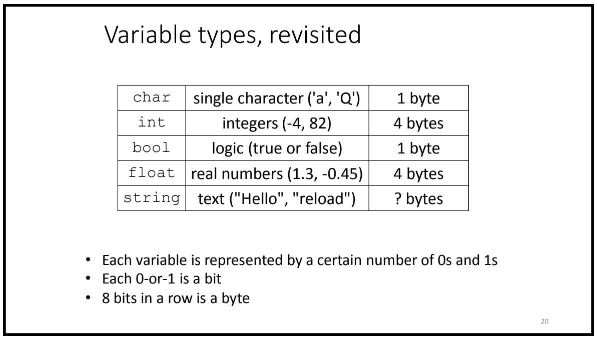
00110000=?

10010010=?

From numbers to symbols: the ASCII table



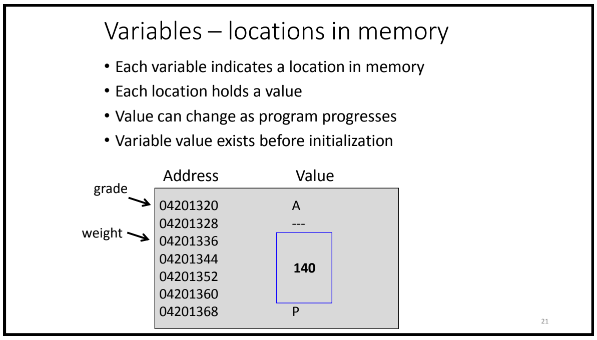
Numeric code	Character	Numeric code	Character	Numeric code	Character	Numeric code	Character
45	-	65	A	85	U	105	i
46	.	66	B	86	V	106	j
47	/	67	C	87	W	107	k
48	0	68	D	88	X	108	l
49	1	69	E	89	Y	109	m
50	2	70	F	90	Z	110	n
51	3	71	G	91	[	111	o
52	4	72	H	92	\	112	p
53	5	73	I	93	]	113	q
54	6	74	J	94	^	114	r
55	7	75	K	95	_	115	s
56	8	76	L	96	`	116	t
57	9	77	M	97	a	117	u
58	:	78	N	98	b	118	v
59	;	79	O	99	c	119	w
60	<	80	P	100	d	120	x
61	=	81	Q	101	e	121	y
62	>	82	R	102	f	122	z
63	?	83	S	103	g	123	{
64	@	84	T	104	h	124	|



Variable types, revisited

`char`	single character ('a', 'Q')	1 byte
`int`	integers (-4, 82)	4 bytes
`bool`	logic (true or false)	1 byte
`float`	real numbers (1.3, -0.45)	4 bytes
`string`	text ("Hello", "reload")	? bytes

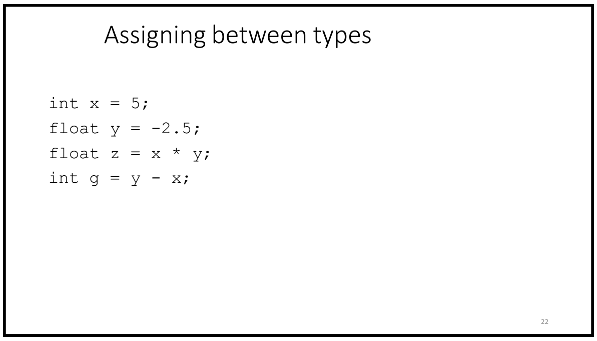
- Each variable is represented by a certain number of 0s and 1s
- Each 0-or-1 is a bit
- 8 bits in a row is a byte



Variables – locations in memory

- Each variable indicates a location in memory
- Each location holds a value
- Value can change as program progresses
- Variable value exists before initialization

	Address	Value
grade →	04201320	A
	04201328	---
weight →	04201336	**140**
	04201344	
	04201352	
	04201360	
	04201368	P



Assigning between types

```
int x = 5;
float y = -2.5;
float z = x * y;
int g = y - x;
```