

WELCOME TO CS 16!

Problem Solving with Computers-I

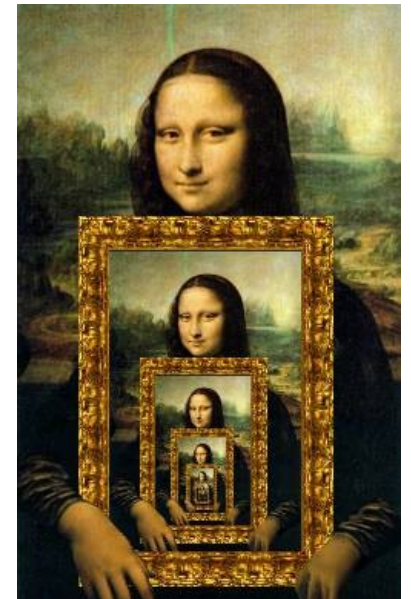


C++

```

#include <iostream>
using namespace std;

int main(){
  cout<<"Hola Facebook\n";
  return 0;
}
  
```



Instructor/TAs

- Lawton Nichols
 - I'm just a PhD Student, so you don't have to call me "professor"
- Office hours:
 - Tuesday, Thursday: 3:30pm–4:30pm in the TA Trailer
 - Or by appointment
- TAs:
 - Adam Ibrahim—Monday 11:45–1:45 in the TA Trailer
 - Kun Wan—Tuesday 10:00–12:00 (possibly Phelps 3525?)
- More info on the course website

Clickers out – frequency AB

About you...

What is your major?

- A. Computer Science
- B. Other College of Engineering
- C. Other²

About you...

What is your past programming experience?

- A. Have never programmed.
- B. Have programmed before “just for fun”
- C. Have taken an introductory CS course
- D. I have a lot of programming experience

About you...

What is your familiarity/confidence with programming in C++?

- A. Know nothing or almost nothing about it.
- B. Used it a little, beginner level.
- C. Some expertise, lots of gaps though.
- D. Lots of expertise, a few gaps.
- E. Know too much; I have no life.

About you...

What is your familiarity/confidence with using version control with Subversion, Git or any other VCS?

- A. Know nothing or almost nothing about it.
- B. Used it a little, beginner level.
- C. Some expertise, lots of gaps though.
- D. Lots of expertise, a few gaps.
- E. Know too much; I have no life.

Clickers, Peer Instruction, and PI Groups

- Find 1-2 students sitting near you. If you don't have any move.
- Introduce yourself.
- This is your initial PI group (at least for today)

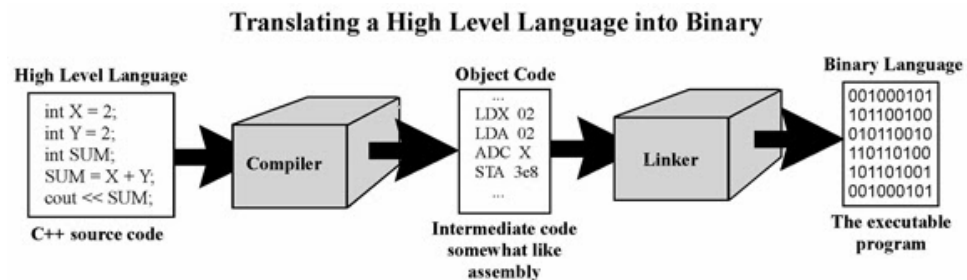
About this course

C++

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

int main(){
    cout<<"Hola Facebook!n";
    return 0;
}
```

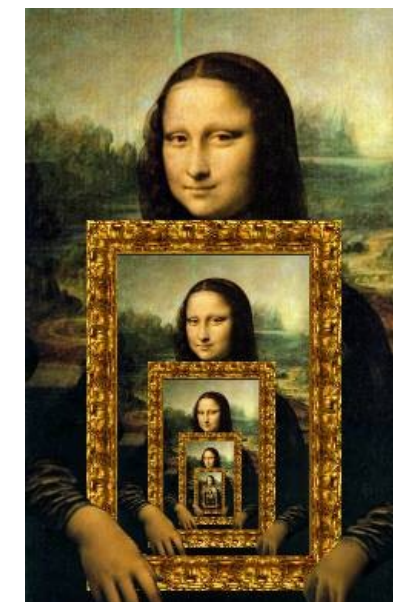
Under the hood of programs



GitHub



Solve fun problems!



Why learn C++?

(Discuss with your group)

C++

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

int main(){
    cout<<"Hola Facebook!\n";
    return 0;
}
```

Which of these reasons is the most important reason to you?

Why learn what goes on under the hood of programs?

(Discuss with your group)

Which of these reasons is the most important reason to you?

Why learn github?

(Discuss with your group)

Which of these reasons is the most important reason to you?

iClickers: You must bring them

- Buy an iClicker at the Bookstore
- Register it on GauchoSpace

Assigned Reading from

- Problem Solving with C++, Walter Savitch, Edition 9

You must **attend** class and lab sections

You must **prepare** for class

You must **participate** in class

Course Logistics

- Grading

- Class and section participation (iClickers): : 2%
 - Homeworks : 13%
 - Lab (programming) Assignments : 35%
 - Midterm exams: (two, 15% each) : 30%
 - Final exam : 20%
-
- Less than 75% iClicker response \equiv missing a class/section
 - No makeups for exams. Make sure you have no scheduling conflicts with exams
 - No LATE submissions unless you have a real emergency!

Course website!

<https://ucsb-cs16-su17.github.io/>

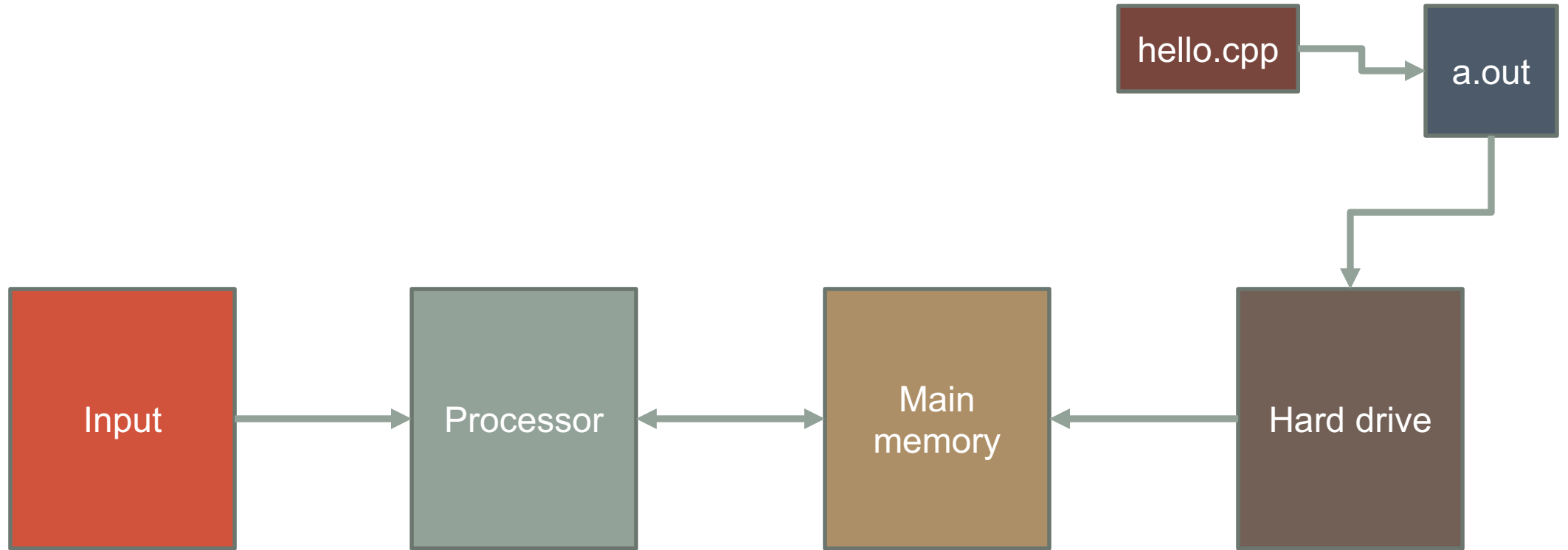
* ATTENDANCE in sections and lecture is REQUIRED

Assignment Calendar

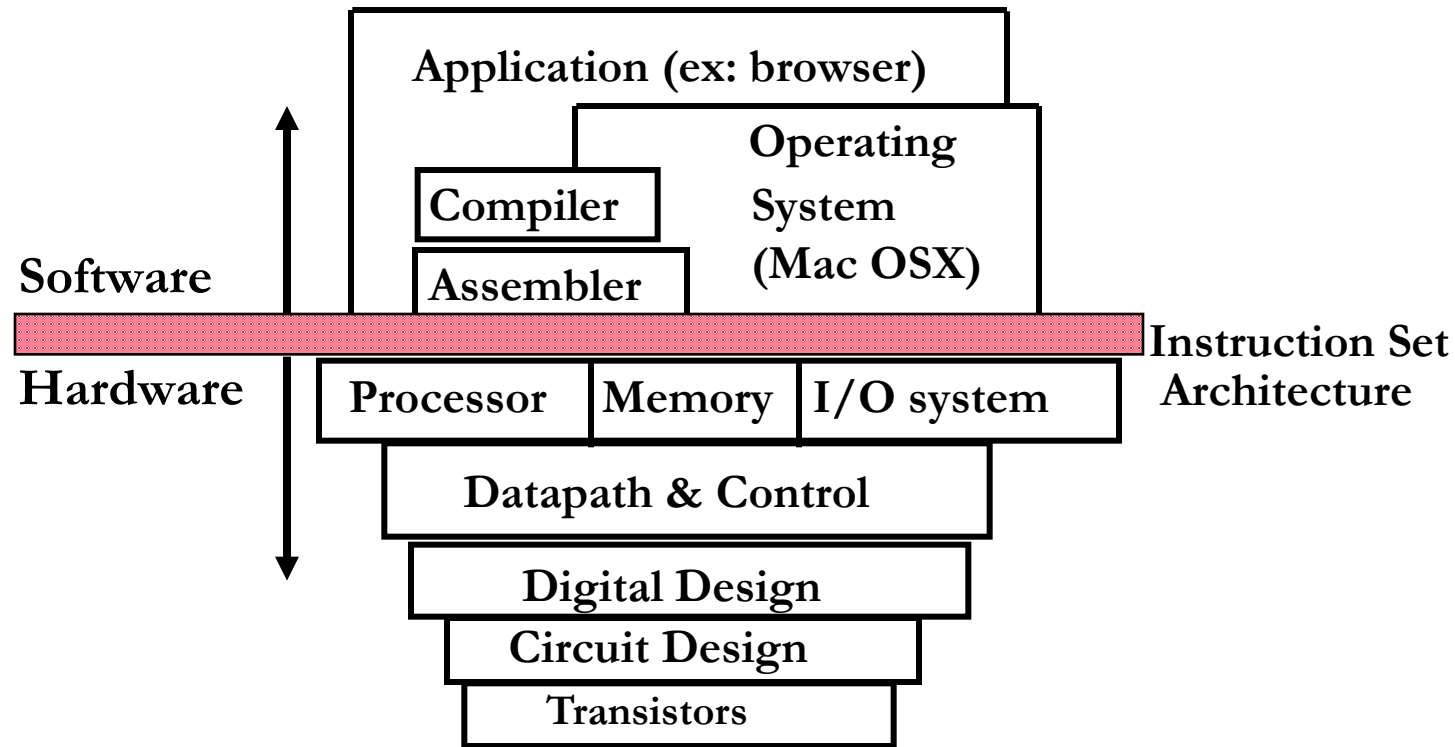
Week	S	M	T	W	R	F	S
1	04/02	04/03	04/04	04/05	04/06	04/07	04/08
		h01 assigned h02 assigned lect01 : Course overview, a gentle intro to C++ - Standard I/O, variables, if-else control structure First day of classes	lab00 assigned	lect02 : Evaluating C++ expressions, simple flow control- for, while loops, nested and multi-way if-else			
2	04/09	04/10	04/11	04/12	04/13	04/14	04/15
		h01 due 02:00pm h02 due 02:00pm h03 assigned h04 assigned lect03 : Nested loops, git, intro to lab01	lab00 due 11:59am lab01 assigned	lect04 : C++ functions and function call mechanics, passing parameters to programs			

- For more information, see our Assignment Calendar: <https://ucsb-cs16-su17.github.io/info/calendar/>
- All sections will be in PHELPS 3525
- Open labs: CSIL in Harold Frank Hall
- The schedule for sections, office hours and open lab hours is available on our class Google Calendar: <https://ucsb-cs16-su17.github.io/info/schedule/>

Basic components of a computer

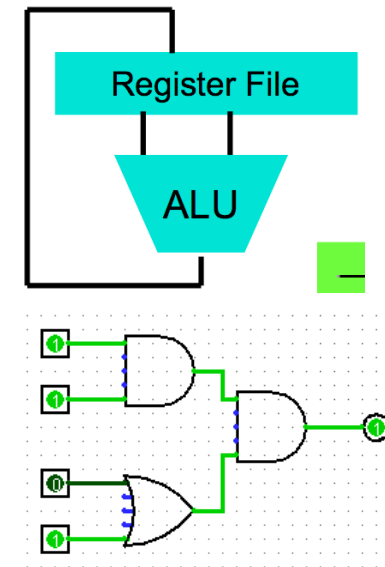


How do we handle complexity?

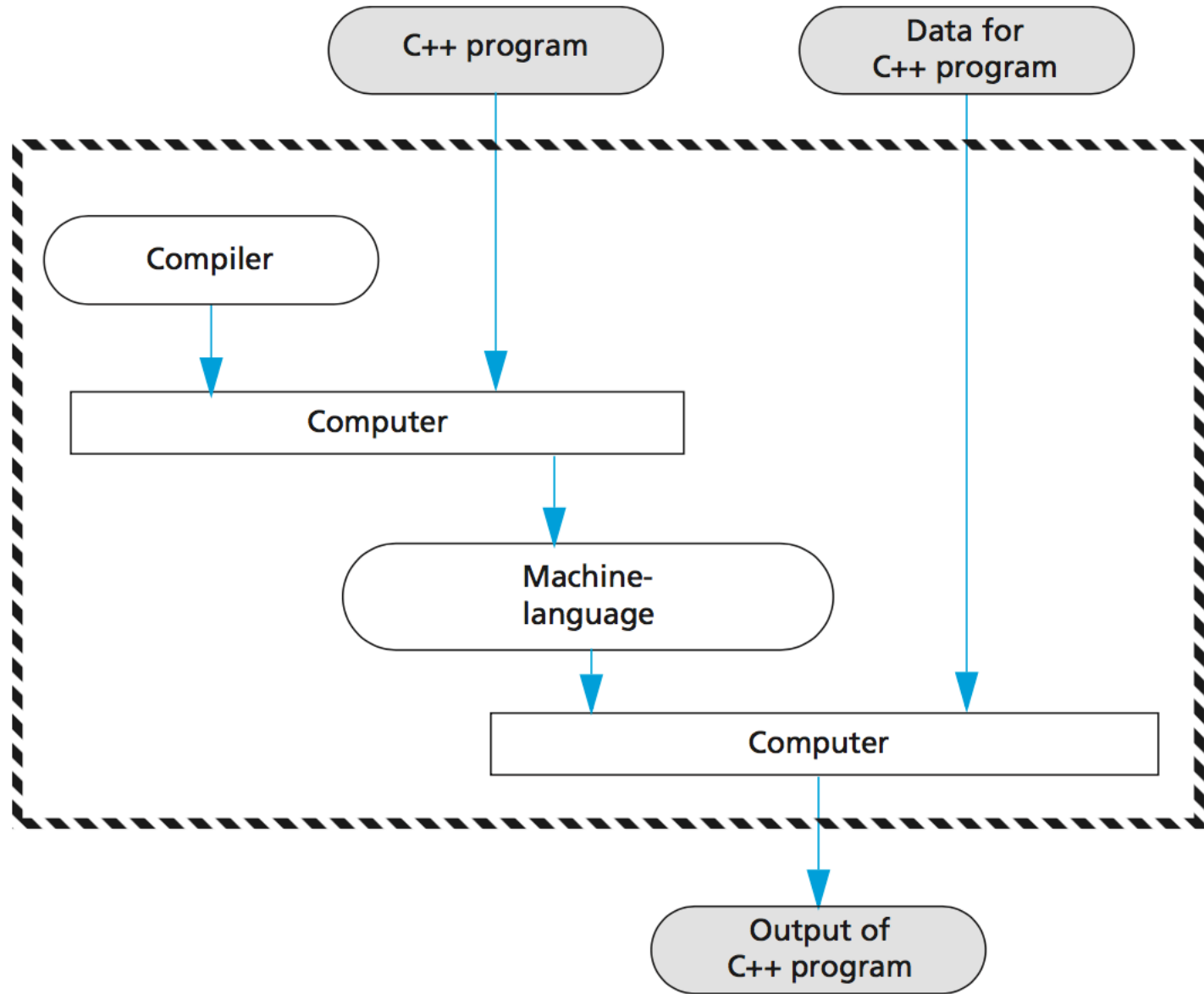


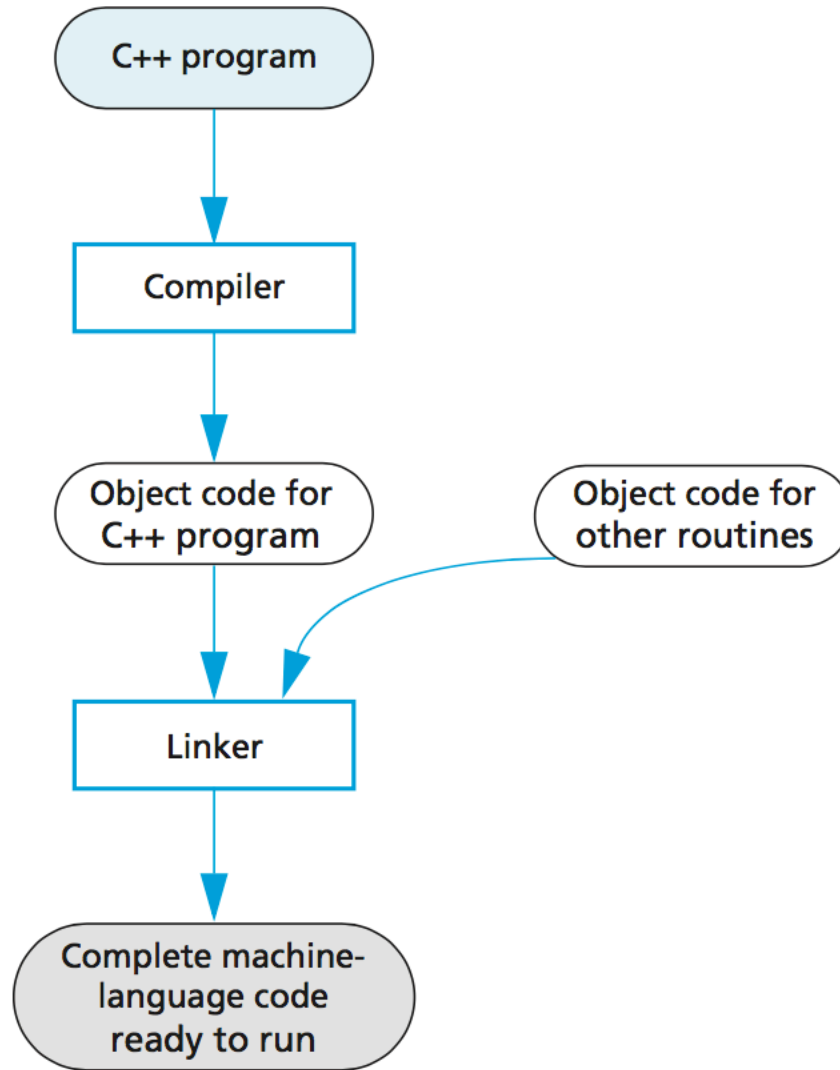
```
temp = v[k];  
v[k] = v[k+1];  
v[k+1] = temp;  
ldr r0, [r2]  
ldr r1, [r2, #4]  
str r1, [r2]  
str r0, [r2, #4]
```

```
0000 1001 1100 0110 1010 1111 0101 1000  
1010 1111 0101 1000 0000 1001 1100 0110  
1100 0110 1010 1111 0101 1000 0000 1001  
0101 1000 0000 1001 1100 0110 1010 1111
```



- Big idea: Coordination of many *levels of abstraction*





Which code produces a compile-time error?

A.

```
int main(){  
    cout<<"Enter two numbers:";  
    cin>>a >> b;  
    cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;  
    return 0;  
}
```

B.

```
int main(){  
    int a, b;  
    cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;  
    return 0;  
}
```

C.

Both **A** and **B**

D.

Neither **A** or **B**

C++ Variables and Datatypes

- **Variables** are containers to store data
- **C++** variables must be “declared” before they are used by specifying a datatype
 - `int`: Integers
 - `double`: floating point numbers
 - `char`: characters

```
int main() {  
  
    cout<<"Enter two numbers:";  
    cin>>a >> b;  
    cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;  
}
```

Will the above code work?

C++ Uninitialized Variables

- Value of uninitialized variables is “undefined”
- Undefined means “anything goes”
- Can be a source of tricky bugs
- What is the output of the code below?

```
int main() {  
    int a, b;  
    cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;  
}
```

Variable Assignment

- The values of variables can be initialized...

```
int myVariable = 0;
```

-or-

```
int myVariable;  
myVariable = 0;
```


Variable Assignment

- ...or changed on the fly...

```
int myVariable = 0;  
myVariable = 5 + 2;
```

Variable Assignment

- ...or even be used to update the same variable!

```
int myVariable = 0;  
myVariable = 5 + 2;  
myVariable = 10 - myVariable;  
myVariable = myVariable==0;
```

What happens when we execute the last statement?!

Control flow: if statement

- The `condition` is a **Boolean expression**
- These can use relational operators

```
if ( 1 < 2 ) {  
    cout<< "foo" ;  
}  
  
if ( 2 == 3 ) {  
    cout<<"foo" ;  
}  
  
if ( Boolean expression) {  
    // statement 1;  
    // statement 2;  
}
```

C++ DATA TYPES

BASIC CONTROL FLOW

Problem Solving with Computers-I
Chapter 1 and Chapter 2

C++

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

int main(){
    cout<<"Hola Facebook!n";
    return 0;
}
```



CLICKERS OUT – FREQUENCY AB

Program compilation

What does it mean to “compile” a C++ program?

- A. Write the implementation of the program in a .cpp file
- B. Convert the program into a form understandable by the processor
- C. Execute the program to get an output
- D. None of the above

Kinds of errors

Which of the following types of errors is produced if our program divides a number by 0?

- A. Compile-time error
- B. Run-time error
- C. Both A and B
- D. Neither A or B

Review: Which code produces a compile-time error?

A.

```
int main(){  
    cout<<"Enter two numbers:";  
    cin>>a >> b;  
    cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;  
    return 0;  
}
```

B.

```
int main(){  
    int a, b;  
    cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;  
    return 0;  
}
```

C.

Both **A** and **B**

D.

Neither **A** or **B**

C++ Variables and Datatypes

- **Variables** are containers to store data
- **C++** variables must be “declared” before they are used by specifying a datatype
 - `int`: Integers
 - `double`: floating point numbers
 - `char`: characters
 - `string/char*/char[]`: strings

Type Name	Memory Used	Size Range	Precision
<i>short</i> (also called <i>short int</i>)	2 bytes	-32,768 to 32,767	(not applicable)
<i>int</i>	4 bytes	-2,147,483,648 to 2,147,483,647	(not applicable)
<i>long</i> (also called <i>long int</i>)	4 bytes	-2,147,483,648 to 2,147,483,647	(not applicable)
<i>float</i>	4 bytes	approximately 10^{-38} to 10^{38}	7 digits
<i>double</i>	8 bytes	approximately 10^{-308} to 10^{308}	15 digits
<i>long double</i>	10 bytes	approximately 10^{-4932} to 10^{4932}	19 digits

C++ Uninitialized Variables

- Value of uninitialized variables is “undefined”
- Undefined means “anything goes”
- Can be a source of tricky bugs
- What is the output of the code below?

```
int main() {  
    int a, b;  
    cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;  
}
```

Variable Assignment

- The values of variables can be initialized...

```
int myVariable = 0;
```

-or-

```
int myVariable;  
myVariable = 0;
```

- ...or changed on the fly...

```
int myVariable = 0;  
myVariable = 5 + 2;
```

Variable Assignment

- ...or even be used to update the same variable!

```
int myVariable = 0;  
myVariable = 5 + 2;  
myVariable = 10 - myVariable;  
myVariable = myVariable==0;
```

Control Flow: if-else

```
if (x > 0) {  
    pet = dog;  
    count++;  
} else {  
    pet = cat;  
    count++;  
}
```

- Can you write this code in a more compact way?

Let's play Fizzbuzz

Let's code Fizzbuzz -1.0

\$ Enter a number: 1

1

\$ Enter a number: 2

2

\$ Enter a number: 3

fizz

\$ Enter a number: 4

4

\$Enter a number: 5

5

\$Enter a number: 6

fizz

\$Enter a number: 7

7

\$Enter a number: 15

fizz

Fill in the 'if' condition to detect numbers divisible by 3

A. `x / 3 == 0`

B. `!(x % 3)`

C. `x % 3 == 0`

D. Either B or C

E. None of the above

```
if ( _____ )  
    cout << x << "is divisible by 3 \n" ;  
}
```