C++ DATA TYPES
BASIC CONTROL FLOW

Problem Solving with Computers-I

Chapter 1 and Chapter 2

https://ucsb-cs16-wi17.github.io/

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

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

GitHub

CLICKERS OUT – FREQUENCY AB

Review: 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

Review: 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**

Review: 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

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;
```

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
```

Control flow: if statement

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

```
if ( Boolean expression) {
  // statement 1;
  // statement 2;
}
```

- In C++ 0 evaluates to a false
- Everything else evaluates to true

Examples of if statements

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

```
if ( 1 < 2 ) {
   cout<< “foo” ;
}

if ( 2 == 3) {
   cout<<“foo” ;
}
```

Use the curly braces even if you have a single statement in your if

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” ;
 }
```

Control Flow: if-else

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

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

Control Flow: Multiway if-else

```
if (x > 100){
   pet = dog;
   count++;
}  else if (x > 90){
   pet = cat;
   count++;
}  else {
   pet = owl;
}
```

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

Let’s code Fizzbuzz -2.0

```
$ Enter a number:  1
1
$ Enter a number:  2
2
$ Enter a number:  3
fizz
$ Enter a number:  4
4
```

```
$Enter a number:  5
buzz
$Enter a number:  6
fizz
$Enter a number:  10
buzz
$Enter a number:  15
fizzbuzz
```

Control Flow: for loops

```
int x;

for ( x = 0; x < 50; x++ ){
   cout << x ;
}
```

Fizzbuzz – 3.0

```
1
2
fizz
4
buzz
fizz
7
8
fizz
buzz
fizzbuzz
```

Let’s code Fizzbuzz 2.0!

Write a program that calculates the sum

1+2+3+4+ …...+100

What is printed by this code?

A. Inside if

B. Outside if

```
int myVar =0;


if (myVar == 0)
    cout<<"inside if\n“;

cout<<"outside if \n“;
```

What is printed by this code?

A. Inside if

B. Outside if

```
int myVar =0;


if (myVar = 0)
    cout<<"inside if\n“;

cout<<"outside if \n“;
```

Next time

- While loops, nested loops
- Github