
C++ DATA TYPES BASIC CONTROL FLOW

Problem Solving with Computers-I
Chapter 1 and Chapter 2

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

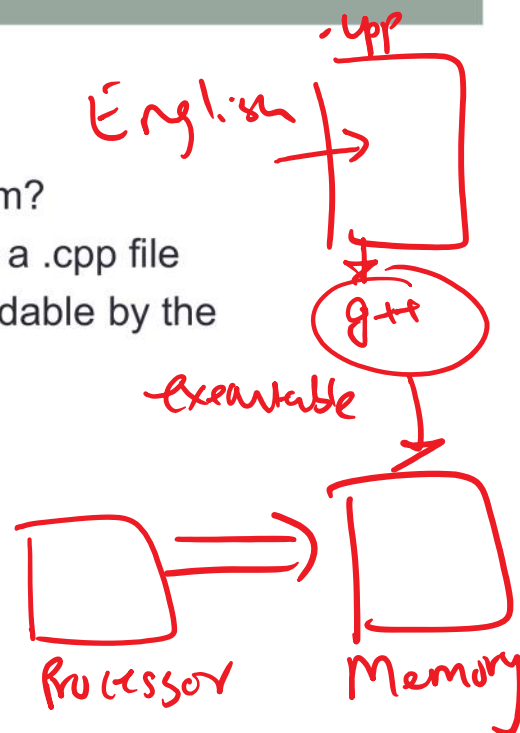


CLICKERS OUT – FREQUENCY AB

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

cout << "Hello";

Syntax error (Violated C++ Grammar)

Logic error → a = b / 0 ;

⇒ 0101010110101
↑
Instruction
DIV A, B, 0

Review: Which code produces a compile-time error?

- A.** *return type*
`int main(){
 cout<<"Enter two numbers:";
 cin>>a >> b;
 cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;
 return 0;
}`
a & b were not declared → int a; *a* ?
- B.** `int main(){
 int a, b;
 cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;
 return 0;
}`
int a; } → int a, b;
- 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

data type

int pet ;

string pet = "Dog";

char pet = 'a';

'Dog'; ← string

'Dog'; X not a string in C++

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...

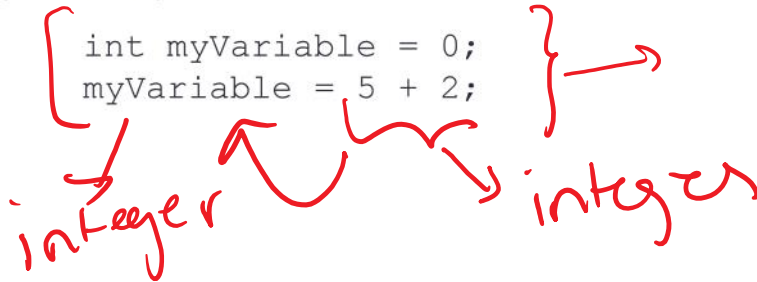
```
int myVariable = 0;
```

-or-

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

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

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



Type mismatch
myVariable = "dog";
(int) (string)

myVariable



Variable Assignment

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

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

//value of myVariable is 0 (false)

myVariable → false → 0

10 - 7 because == higher precedence than assignment(=)

~~10 * 30~~

myVariable = (myVariable == 0);

OR
a = ((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** → expression that evaluates to either a true or false
- 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" ;  
}
```

→ true

foo is printed

```
if ( 2 == 3 ) {  
    cout<<"foo" ;  
}
```

→ false

foo is not printed

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" ;  
}
```

$!(0) \rightarrow 1$

Logical NOT operator
 $! \text{ true} \rightarrow \text{false}$
 $! \text{ false} \rightarrow \text{true}$
 $! - 1 = 0$

Control Flow: if-else

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

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

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

Control Flow: Multiway if-else

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

- 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
```