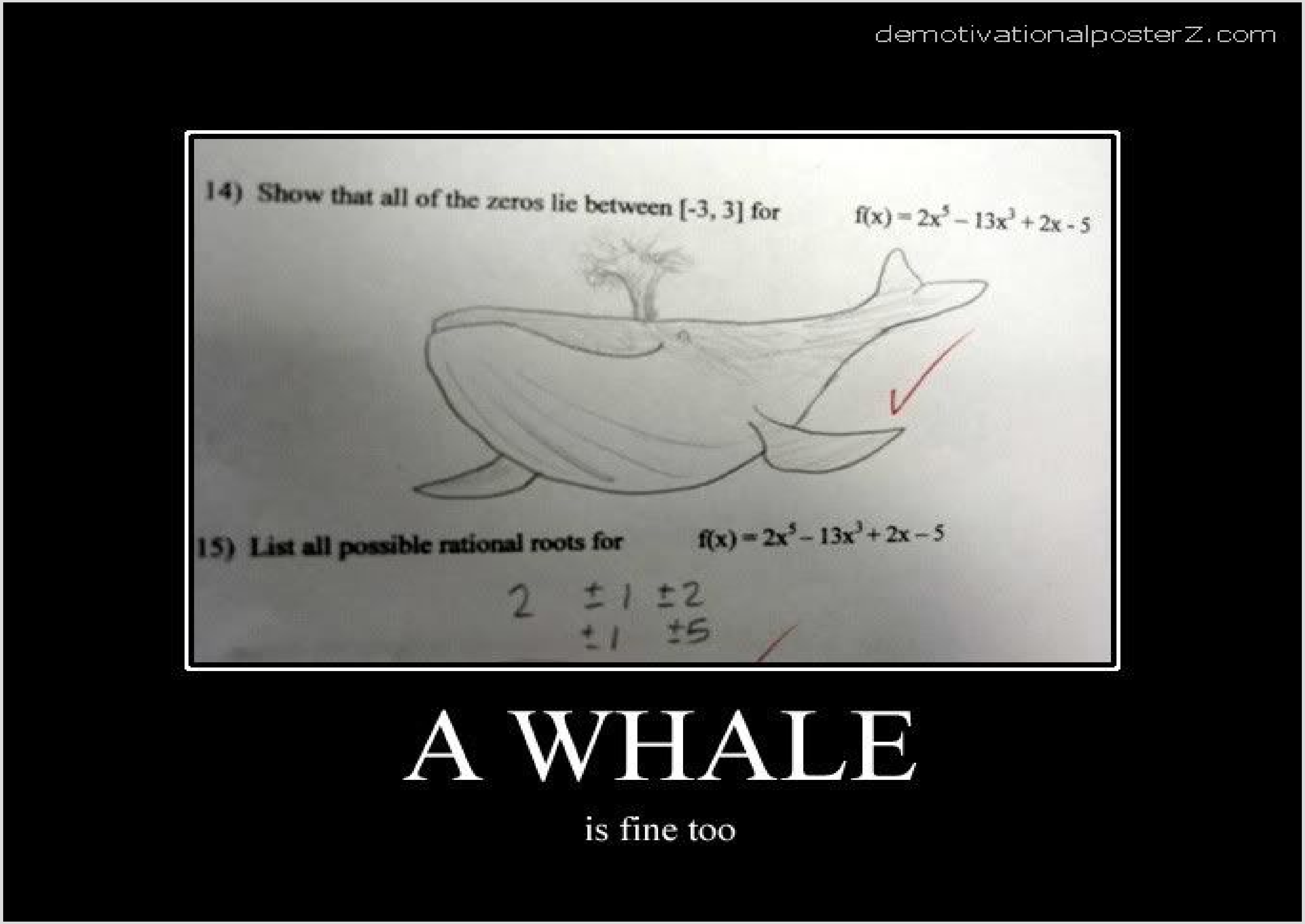
Midterm 2 Review



14) Show that all of the zeros lie between [-3, 3] for $f(x) = 2x^5 - 13x^3 + 2x - 5$

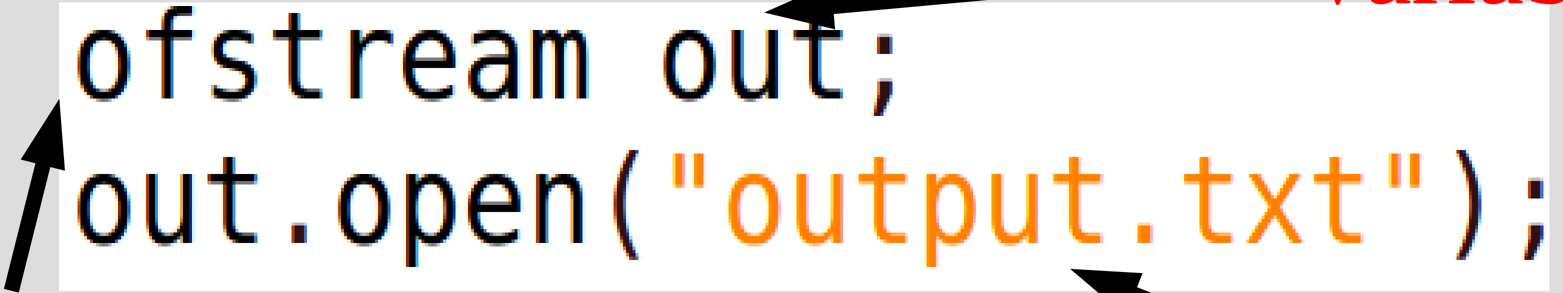
15) List all possible rational roots for $f(x) = 2x^5 - 13x^3 + 2x - 5$

2 ±1 ±2
±1 ±5

File I/O

For files you must first open them:

Variable name



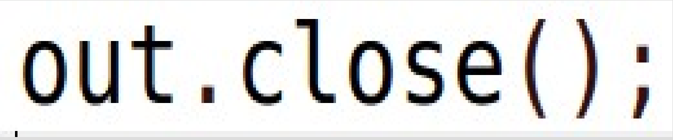
```
ofstream out;
out.open("output.txt");
```

Type

File name

Then you use “out” instead of “cout” or “cin” depending on if it is an ostream of istream

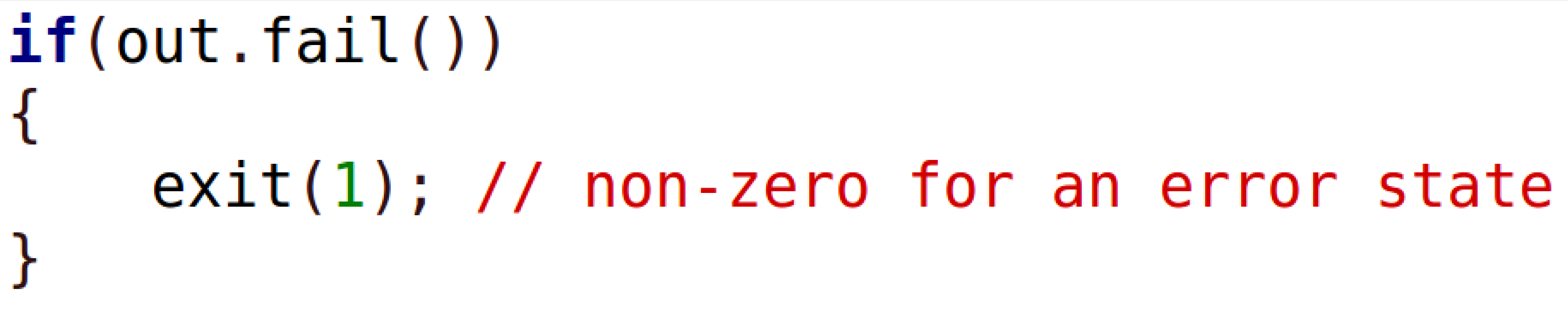
Also close when done:



```
out.close();
```

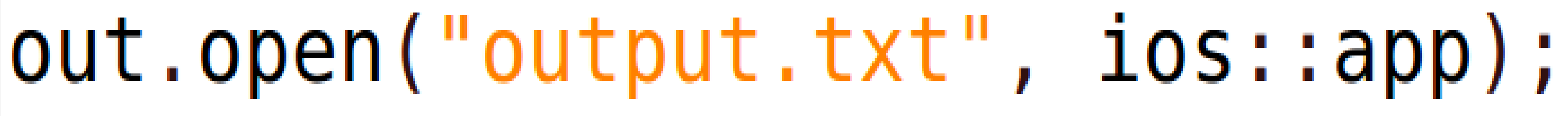
File I/O

Can check to see if the program is correctly sending/receiving to/from file:



```
if(out.fail())
{
    exit(1); // non-zero for an error state
}
```

If you want to add to the file instead of replacing it, you have to specify when opening

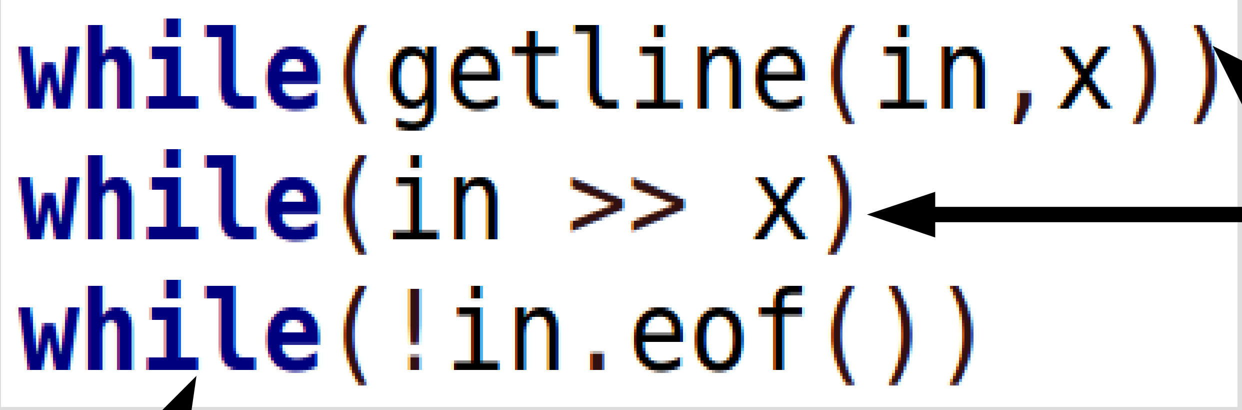


```
out.open("output.txt", ios::app);
```

End of file (EOF)

When there is nothing left in a file to read, we call it end of file

C++ is fairly nice about handling EOF, and you can detect it in 3 ways:



```
while(getline(in,x))
while(in >> x)
while(!in.eof())
```

reads from file

does not read from file (just tells if at end)

File I/O

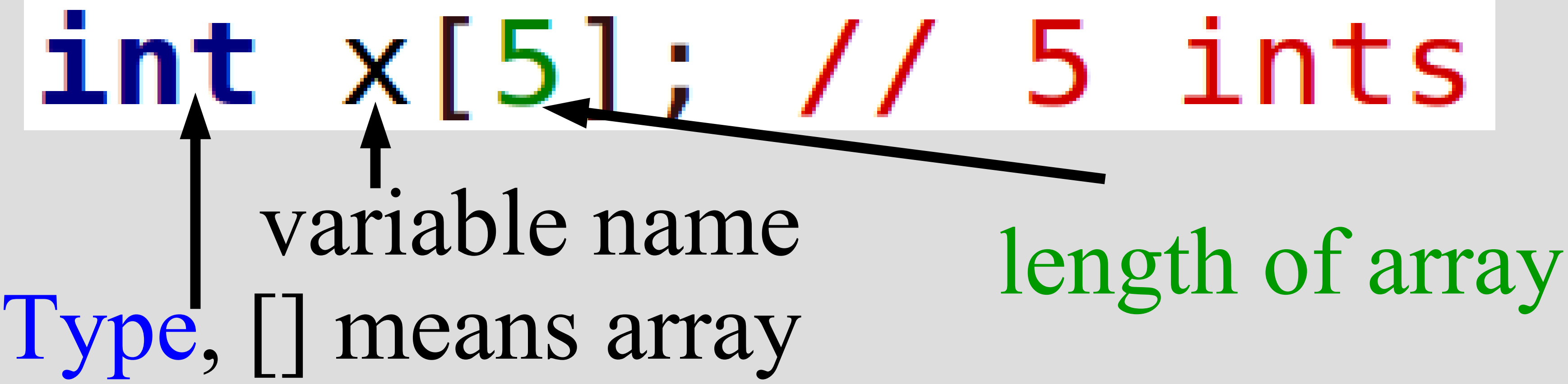
Q: Read all the numbers from “numbers.txt” and put their sum in “sum.txt”

If you cannot read “numbers.txt”, put “NaN” into “sum.txt”
(you can get this by doing 0.0/0.0)
(technically the above is -NaN...)

(see: fileQ.cpp)

Arrays

Arrays store multiple things of the same type



```
int x[5]; // 5 ints
```

variable name

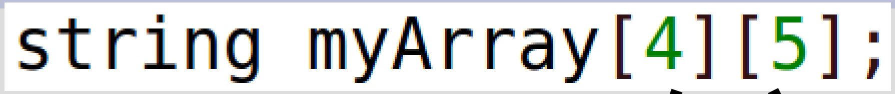
length of array

Type, [] means array

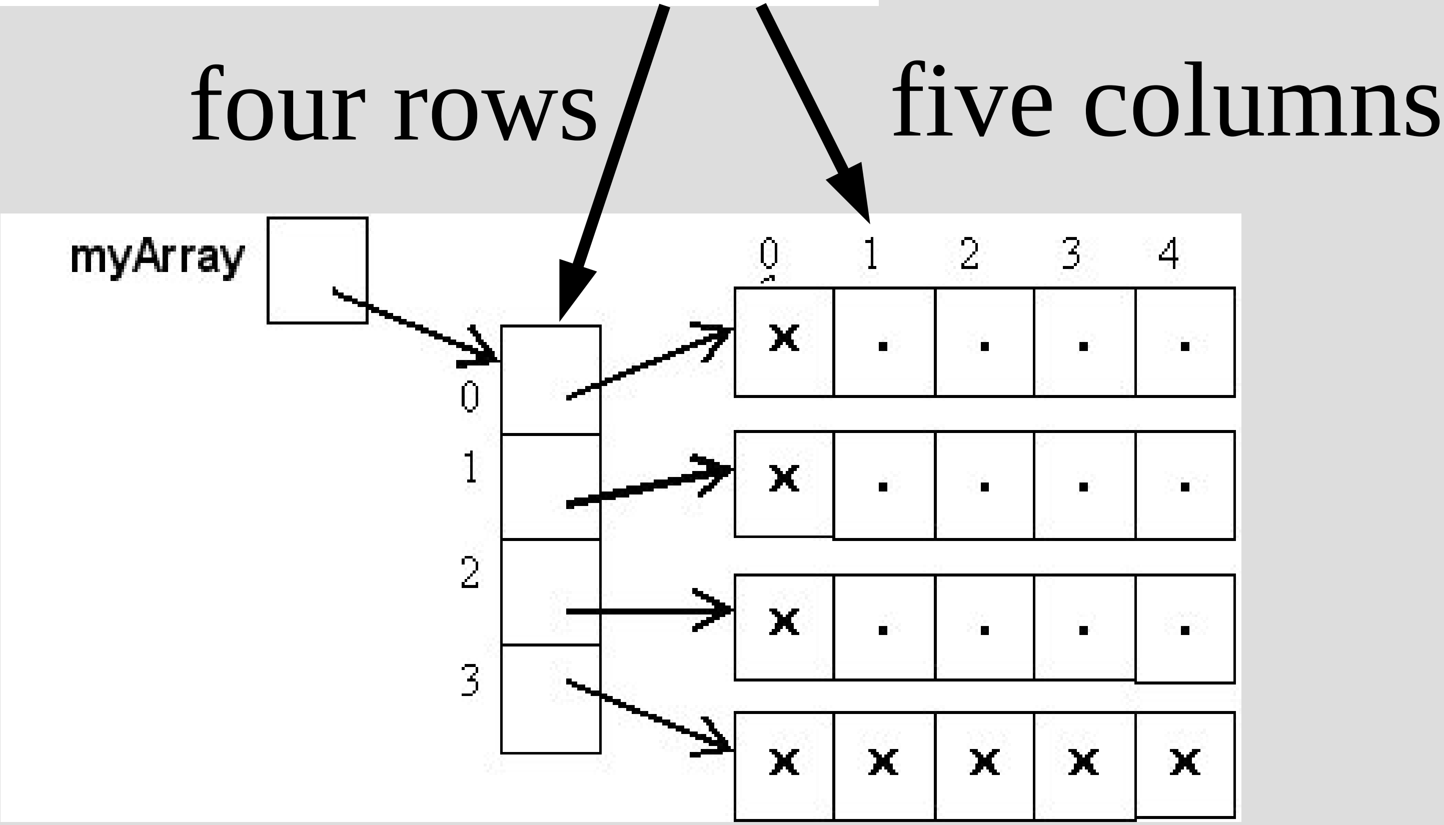
After declaration **any use of [ ]** is interpreted as element indexing

Arrays are memory addresses, shares with functions (cannot call-by-reference)

Multidimensional Arrays



```
string myArray[4][5];
```



four rows

five columns

Must specify (some parts of) size when using as argument in function (all but first)

Arrays

Q: Write a function that takes two int arrays of length 11 as input. Return true if the first array has more larger numbers when compared to the second element by element:

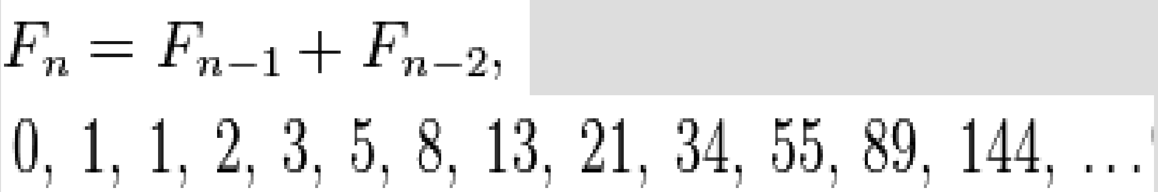
first = [1, 2, 3, 4], second = [90, 0, 0, 0],
then function would return true as first array has 3 larger elements and 1 smaller:
1 < 90, 2 > 0, 3 > 0, 4 > 0
(see: arrayQ.cpp)

Recursion

There are two important parts of recursion:

- -A stopping case that ends the recursion
- -A reduction case that reduces the problem

Identify the problem sub-structure, then move inputs towards the base case



$$F_n = F_{n-1} + F_{n-2},$$

$$0,\ 1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34,\ 55,\ 89,\ 144,\ \ldots$$

You can assume your function works as you want it to (and it will if you do it properly!)

Recursion

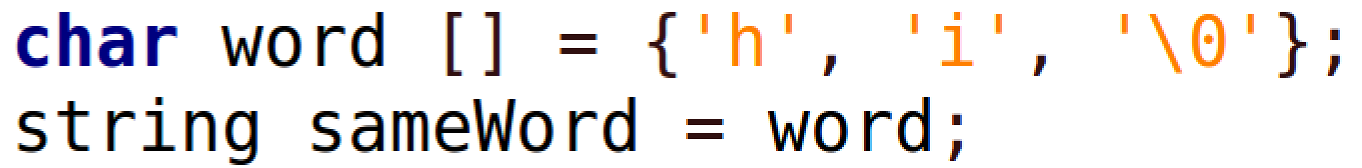
Q: Write a recursive function that keeps asking if the user wants to stop.
When the character 'q' is pressed, stop and **return** how many inputs other than q they entered

Example input: aabeq
Example output: 4 other inputs

(see: recursionQ.cpp)

C-Strings and strings

c-string uses null character to tell when to end



```
char word [] = {'h', 'i', '\0'};
string sameWord = word;
```

(c++) string is a class (which is a type) and is newer and has many functions:
- find(), substr(), at() or [ ], etc.

Essential for dealing with more than one char at a time

C-Strings and strings

Q: Write a function that takes a c-string (char array) as input (and its length) and changes it to display half as much when couted (i.e. “cookies” -> “cook” or “coo”)

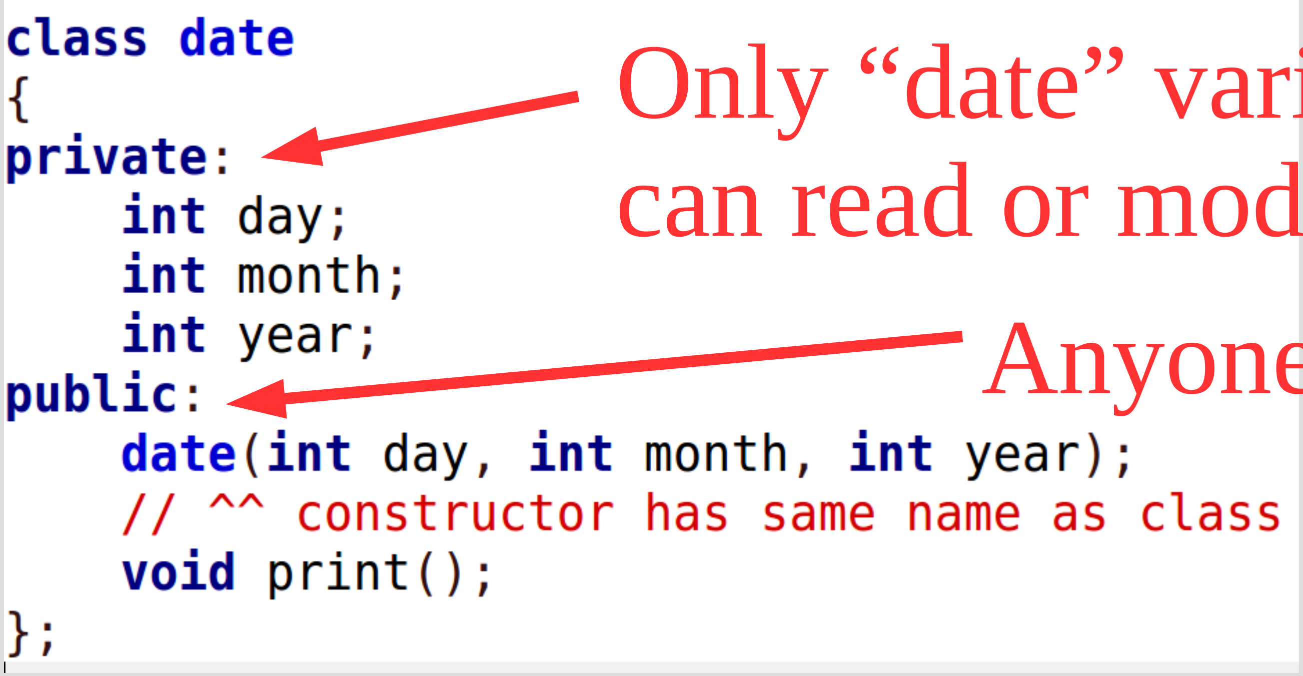
(see: cstringQ.cpp)

Q: Make a word game that repeatedly reads in words until the user repeats a word they have already entered. At this point tell the user they have lost

(see: wordGame.cpp)

Classes

A class is a way to bundle functions and variables (different types) into one logical unit



```
class date
{
private:
    int day;
    int month;
    int year;
public:
    date(int day, int month, int year);
    // ^^ constructor has same name as class
    void print();
};
```

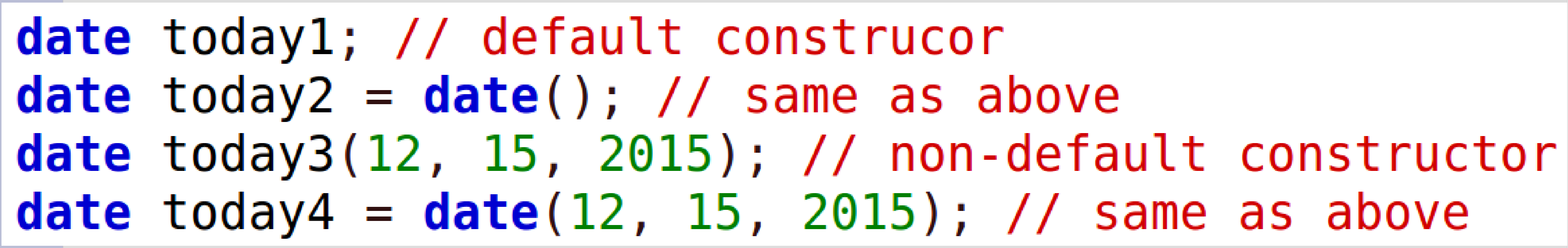
Only “date” variables can read or modify

Anyone can edit/use

Classes are custom made types (like int), that you make and define

Classes

Every time you actually create an object of the class type, you must run a constructor



```
date today1; // default construcor
date today2 = date(); // same as above
date today3(12, 15, 2015); // non-default constructor
date today4 = date(12, 15, 2015); // same as above
```

Constructors should initialize (probably) all variables inside the class

Classes

Suppose you were going to code up what your class schedule looks like

You take multiple classes, but each class has a: name, section, lecture time, and credits
Q:
Make a class that would store all the student's classes for this semester (just the definition)
Then write a function to determine how many credits they are taking (see: credits.cpp)