Example of a C++ Class

From Tom Anderson's "A Quick Introduction to C++"

Stack Class

```cpp
class Stack {
public:
    void Push(int value); // Push it, check for overflow
    int top;              // Index of stack top
    int stack[10];       // Elements of stack
};

void Stack::Push(int value) {
    ASSERT(top < 10);    // stack shouldn't overflow
    stack[top++] = value;
}
```

Notes

- Class names
  - are `noun` ("Stack")
  - start with upper case
- Member function names
  - are `verbs` ("push")
  - start with upper case
- Variables names
  - start with lower case

This is specific to this example. Not true in general for C++
Usage

```c
main() {
    Stack s0;
    Stack s1;
    s0.Push(21);
    s1.Push(99);
}
```

Why Private?

```c
main() {
    Stack s0;
    Stack s1;
    s0.Push(21);
    s1.Push(99);
    s0.top = 99;       // maybe a bad idea!
}
```

Using Private

```c
#include <assert.h>

class Stack {
public:
    void Push(int value); // Push it, check for overflow
private:
    int top;            // Index of stack top
    int stack[10];     // Elements of stack
};

void Stack::Push(int value) {
    ASSERT(top < 10); // stack shouldn't overflow
    stack[top++] = value;
}
```
Notes

- Class variables should all be private
- "Push" can use "top", but code outside class cannot.

```
main() {
    Stack s0;
    s0.Push(21);
    s0.top = 99; // Now a compilation error!
}
```

Constructors

```
class Stack {
  public:
    Stack(int sz); // Constructor: init vars, allocate space
    void Push(int value);
  private:
    int size;       // Max capacity of stack
    int top;        // Index of the lowest unused position.
    int* stack;     // Pointer to array that holds contents.
};

Stack::Stack(int sz) {
  size = sz;
  top = 0;
  stack = new int[size]; // let's get an array of integers.
}
```

Notes

- No return type for constructor
- "new" used inside constructor to allocate dynamic array