Today’s lecture

• C to C++
C recap

• Useful features of C
  – CPP
  – structure, typedef
  – memory management
  – system calls
  – recursion

• C is also the basis for C++ ............
C++

- Object Oriented - Allows you to build/compose very complex applications from building blocks
- Appeared around 1984 (Bjarne Stroustrup, Bell Labs)
- ANSI standard 1997
- Syntax is like C. Getting started: a few extra keywords + few new formalized concepts. One lecture – honest!!!
- Book “C++ The Core Language” – O’Reilly
- Successful because you can compose applications from other peoples building blocks. Windows etc.…
- Very complex in detail, like Mathematica takes many years to learn everything!!
C++ concept

• C language + **classes**
• Class is a formal way to think about good program design.
  – Modularity, encapsulation, hierarchy, abstraction
• A class has
  – Methods ( program logic)
  – Data ( variables )
  – can be private or public
• Example “string”
  – Methods: set, get
  – Data: string text, string length
C++ Basic Example

main()
{
    String s;
    printf("Executable code starting\n");
    s.set("Hello");
    printf("%s\n",s.get());
    printf("Executable code ending\n");
}

Compile using g++
Will write out hello + some other stuff
C++ Basic Example

```c++
main()
{
    String s;
    printf("Executable code starting\n");
    s.set("Hello");
    printf("%s\n", s.get());
    printf("Executable code ending\n");
}
```
/* ===== Class interface definition ===== */
class String {
public:
    String();                    /* Constructor           */
    ~String();                  /* Destructor            */
    void set(char *s);      /* Set a string          */
    char *get(); /* Get string value      */
private:
    char *str;                  /* Pointer to the string */
    int lngth;                 /* Length of the string  */
};
String Class – Example Methods

/* Set str to point to a private copy of s */
void String::set(char *s) {
    lngth = strlen(s);
    str = new char[lngth+1];
    strcpy(str, s);
}

/* Return the pointer to the string */
char *String::get() {
    return str;
}

10/3/02  12.010 Lec 09  8
String Class – Example Methods

/* Constructor */
String::String() {
    str = 0;
    set(""");
    printf("I created a string\n");
}

/* Destructor */
String::~String() {
    delete[] str;
    printf("I deleted a string\n");
}
Application Example

Throwing a ball in the air

\[ w = w_0 + gt \]

\[ z = z_0 + \int_0^t wdt \]

Get initial velocity and length of “experiment”.

Calculate time evolution of \( w \) and \( z \).

Print out “trajectory”
C “Procedural” Form

main ( )
{
    float t=10.; float w0=10.;
    t_gball *theBall;/* Stats for the ball */

    /* Allocate space for full ball time history */
    createBall(w0, &theBall);
    /* Step forward the ball state */
    stepForwardState( t, &theBall );
    /* Write table of output */
    printTrajectory( t, w0, theBall);
}
C++ Using “Ball” Class

main()
{float w0 = 10.; float t=10.;
   Ball b;
   b.initialize(w0);
   b.simulate(t);
   b.printTrajectory();
}

All info. is held in “b”. Fewer args, cleaner “abstraction”.
Homework

- Questions
- Due next Thursday (10\textsuperscript{th} October)