- A summary of key course policies; e.g. How is in-class participation measured? What is expected to prepare for lab sessions? How are a reasonable number of absenses and late/missing work accomodated for each of the graded components of the course? etc.
- What are the key components of every C program (e.g. an #included library, main(), etc.)? That is, what does a *blank template* for a C program look like?
- How is basic input and basic output handled in C programs? What does the "f" stand for in printf() and scanf()? As an example, how do you read an integer from user-input, and how can your print an integer variable to the console?
- What are the main differences between programming in C vs. C++? What types of applications are best suited for each programming language?

In class participation - done in iClicker
Lab session prep - do the assigned reading and complete an outline to be shown to the TA
Absences and late/missing work - 2 lab absences are acceptable but all labs must be completed even if absent. No late work or make ups for exams. Projects can be a total of 6 days late with an allotted time of 2 days per project. No late zybooks are accepted. Only on time iClickers will be accepted. The lowest 4 scores on iClicker will be dropped.

The key concepts of a program in C are the Header, main function, variable declaration, body, and return statement.

```
#include <stdio.h>

// main function
int main()
{

   // body
   printf("Hello World");

   // return statement
   return 0;
}
```

Some differences between C and C++ are input and output methods. C++ uses std::cin>> and std::cout<< for input/output while C uses scanf() and printf()
And C is a procedural oriented language while C++ also incorporates object-oriented programming which allows a program to be spilt between objects to create less errors.
C is used for low level programs where the main focus is speed and efficiency and is common for operating systems. C++ is used for higher level programming because of its use of objects.