ECE109

ECE109 program outline

```
#include <stdio.h>

/* Your name goes here */

main() {
      eCe109 declarations ;
      eCe109 statements ;
}
```

ECE109 declaration

A single line of the form:

type variable `;`

where *type* can be int or double and *variable* follows the rules in the notes.

ECE109 statements

There are 5 types of statements:

eCe109 input statement
eCe109 output statement
eCe109 arithmetic assignment statement
eCe109 conditional statement
eCe109 while loop

ECE109 input statement

To read into an int variable `i` use

```
scanf("%d", &i) ;
```

To read into a double variable `x` use

```
scanf("%lg", &x) ;
```

eCE109 output statement

Use `printf` as described in Section 11.5.4 of the textbook.

eCE109 arithmetic assignment statement

variable = arithmeticExpression ;

Use the literals described in the class notes and the arithmetic operators described in sections 12.3.3 and 12.3.4 of the textbook. They are the obvious ones.

eCE109 conditional statement

```
if (logicalExpression) {
      eCe109 statements ;
} else if (logicalExpression) {
      eCe109 statements ;
} else {
      eCe109 statements ;
}
```

Use `else` (and `else if`) as needed. Use the relational operators in Table 12.3 and the logical operations in Table 12.4 within the logical expression.

eCE109 while loop

```
while (logicalExpression) {
      eCe109 statements ;
}
```

Again use the relational operators in Table 12.3 and the logical operations in Table 12.4 within the logical expression.

One special example

The following program will add up all the integers in the input stream and print their sum when the input stream has no more input. (You can manually indicate end-of-stream by typing a `^D`.)

```
int sum = 0 ;
int n ;
while (scanf("%d", &n)==1) {
      sum = sum + 1 ;
}
printf("The sum is %d\n", sum) ;
```