Programming in C

Reading Chapter 1

CS300 Data Structures (Fall 2014)

C code tool chain

- Pre-processor   (gcc -E)

- Compiler   (gcc -c)

- Linker (ld, but invoked via gcc for us)

- Loader (ld-linux.so)

CS300 Data Structures (Fall 2014)

Define

- Pointer

- Memory Address

CS300 Data Structures (Fall 2014)

Loops, Arrays, Pointer Review?

```c
1   #include <stdio.h>
2   #include <stdlib.h>
3
4   #define MAX_EXAM_SCORES 50
5   #define MIN_EXAM_SCORES 0
6   #define MAX_SCORE 100
7
8   static void getExamScores (int [], int);
9   static void getExamScore (int *, int);
10  static void printExamScores (const int [], int);
11
12  int main ()
13  {
14    int numExamScores;
15    int *pExamScores;
16
17    printf ("Exam Stats \n\n");
18
19    do
20    {
21      printf ("Enter Number of Exams: ");
22      scanf ("%d", &numExamScores);
23    } while (numExamScores < 0 || numExamScores > MAX_EXAM_SCORES);
```

CS300 Data Structures (Fall 2014)

Loops, Arrays, Pointer Review?

```c
25    pExamScores = (int *) malloc (sizeof (int) * numExamScores);
26
27    if (NULL == pExamScores)
28    {
29      printf ("Error: Cannot Allocate Memory\n\n");
30      exit (EXIT_FAILURE);
31    }
32
33    printf ("\n");
34    getExamScores (pExamScores, numExamScores);
35
36    printf ("\n");
37    printExamScores (pExamScores, numExamScores);
38
39    return 0;
40  }
```

CS300 Data Structures (Fall 2014)

Loops, Arrays, Pointer Review?

```
42  static void getExamScores (int examScores[], int numExamScores)
43  {
44      int i;
45
46      for (i = 0; i < numExamScores; ++i)
47      {
48          getExamScore (&examScores[i], i + 1);
49      }
50  }
51
52  static void getExamScore (int *pExamScore, int whichScore)
53  {
54      do
55      {
56          printf ("Enter Exam Score %d: ", whichScore);
57          scanf ("%d", pExamScore);
58      } while (*pExamScore < 0 || *pExamScore > MAX_SCORE);
59  }
```

Loops, Arrays, Pointer Review?

```
60
61    static void printExamScores (const int examScores[], int numExamScores)
62    {
63       int i;
64
65       printf ("Scores\n");
66
67       for (i = 0; i < numExamScores; ++i)
68       {
69          printf ("%6d\n", examScores[i]);
70       }
71
72       printf ("\n");
73    }
```

CS300 Data Structures (Fall 2014)

Questions on the Previous Program

▶ What is the purpose of the **#define** statement and who does it create work for:

> ▶ The pre-processor
> ▶ The compiler
> ▶ The linker
> ▶ The loader

▶ What is the purpose of using **static** in a function prototype?

▶ Explain scanf

▶ What does "%6d" mean?