

Programming in C

C code tool chain

- Pre-processor (`gcc -E`)
- Compiler (`gcc -c`)
- Linker (`ld`, but invoked via `gcc` for us)
- Loader (`ld-linux.so`)

Example

- ▶ Open a text editor (geany) and type in the following program:

```
#include <stdio.h>

int main (void)
{
    printf ("Hello World");
    return 0;
}
```

- ▶ Save it as (helloworld.c) in a folder called (HelloWorld) in ~/Documents/CS300

Example

- ▶ Now, open a terminal and navigate to the folder HelloWorld that you just created
- ▶ Type: `gcc -c helloworld.c`
- ▶ List the file contents. What file has been created?
- ▶ Type: `gcc helloworld.o`
- ▶ List the file contents. What file has been created?
- ▶ Type: `./a.out`
- ▶ What happens?

Define

- Pointer

- Memory Address

Loops, Arrays, Pointer Review

```
1  /*****
2  File name:    main.c
3  Author:      Computer Science, Pacific University
4  Date:
5  Class:      CS300
6  Assignment:  Exam Stats Shell
7  Purpose:    To show the use of functions in reading and printing exam scores
8              into a dynamically created array.
9  Hours worked: 1
10 *****/
11
12 #include <stdio.h>
13 #include <stdlib.h>
14
15 //*****
16 // Constants
17 //*****
18 #define MAX_NUMBER_OF_EXAM_SCORES 50
19 #define MIN_NUMBER_OF_EXAM_SCORES 0
20 #define MAX_SCORE 100
21 #define SCORES_PER_LINE 4
```

Loops, Arrays, Pointer Review

```
23 //*****
24 // Function Prototypes
25 //*****
26 static void getExamScores (int examScores[], int numExamScores);
27 static void getExamScore (int *pExamScore, int whichScore);
28 static void printExamScores (const int examScores [], int numExamScores);
29
30 /*****
31 Function:    main
32
33 Description: Driver for entering exam scores and and printing them 1 per line
34
35 Parameters:  None
36
37 Returned:   Exit Status
38 *****/
39
40 int main ()
41 {
42     int numExamScores;
43     int *pExamScores;
```

Loops, Arrays, Pointer Review

```
44
45     system ("clear");
46
47     printf ("Exam Stats \n\n");
48
49     do
50     {
51         printf ("Enter Number of Exams: ");
52         scanf ("%d", &numExamScores);
53     } while (numExamScores < MIN_NUMBER_OF_EXAM_SCORES ||
54             numExamScores > MAX_NUMBER_OF_EXAM_SCORES);
55
56     pExamScores = (int *) malloc (sizeof (int) * numExamScores);
57
58     if (NULL == pExamScores)
59     {
60         printf ("Error: Cannot Allocate Memory\n\n");
61         exit(EXIT_FAILURE);
62     }
```


Loops, Arrays, Pointer Review

```
63
64     printf ("\n");
65     getExamScores (pExamScores, numExamScores);
66
67     printf ("Exam Scores\n");
68
69     printExamScores (pExamScores, numExamScores);
70
71     return EXIT_SUCCESS;
72 }
73
```

Loops, Arrays, Pointer Review

```
74  | /*****
75  |  Function:      getExamScore
76  |
77  |  Description:  Gets an exam score and checks if it's between the allowed values.
78  |
79  |  Parameters:   pExamScore - a pointer to an exam score
80  |                whichScore - the number of the scored exam
81  |
82  |  Returned:    None
83  | *****/
84  |
85  | static void getExamScore (int *pExamScore, int whichScore)
86  | {
87  |     do
88  |     {
89  |         printf ("Enter Exam Score %d: ", whichScore );
90  |         scanf ("%d", pExamScore);
91  |     } while (*pExamScore < 0 || *pExamScore > MAX_SCORE);
92  | }
```

Loops, Arrays, Pointer Review

```
94  /*****  
95  Function:    getExamScores  
96  
97  Description: Enters exam scores into an array  
98  
99  Parameters:  examScores    - an array of exam scores  
100              numExamScores - the number of exams scores  
101  
102  Returned:    None  
103  *****/  
104  
105  static void getExamScores (int examScores[], int numExamScores)  
106  {  
107      int i;  
108  
109      for (i = 0; i < numExamScores; ++i)  
110      {  
111          getExamScore (&examScores[i], i + 1);  
112      }  
113  }
```

Loops, Arrays, Pointer Review

```
115  /*****
116  Function:   printExamScores
117
118  Description: Displays the exam scores in the array
119
120  Parameters: examScores   - an array of exam scores
121             numExamScores - the number of exams scored
122
123  Returned:   None
124  *****/
125
126  static void printExamScores (const int examScores [], int numExamScores)
127  {
128      int i;
129
130      for ( i = 0; i < numExamScores; ++i)
131      {
132          printf ("%6d\n", examScores[i]);
133      }
134  }
135
```

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?