

C Assignment One

[illegible]

Diego Trazzi
NWEN241 2015T2

Discuss the four steps of compiling using gcc and give the corresponding gcc options. (You need to specify what the four steps do.)

Gcc goes through a sequence of different intermediate steps before generating final executable. Those intermediate steps are the result of different tools which are invoked internally to complete the compilation of the source code.

The whole Compilation process is broken down into following phases:

- Preprocessing
- Compilation
- Assembly
- Linking

Preprocessor is responsible for text Substitution, Stripping of Comments, and

File Inclusion: the lines in the code that begin with the “#” character are preprocessor directives.

The next stage of the process is the actual compilation of the preprocessed source code to assembly language, for a specific processor (Depending upon the target processor architecture the source code is converted into particular assembly language and it can be known as cross-compilation).

The **assembly stage** converts assembly code into machine code.

The final stage of the compilation process is producing a single executable program file by linking set of object files. An executable file requires many external functions from system and C run-time libraries. The linker will resolve all of these dependencies and plug in the actual address of the functions.

Discuss the major differences between Java and C:

JAVA is Object-Oriented while C is procedural. C breaks down to functions while JAVA breaks down to Objects. C is more procedure-oriented while JAVA is data-oriented.

A second difference between the two is that Java is an Interpreted language while C is a compiled language. C takes code and translates it into something the machine can understand. While with JAVA, the code is first transformed into bytecode. This bytecode is then executed by the JVM(Java Virtual Machine). For the same reason, JAVA code is more portable.

A third difference is C is a low-level language while JAVA is a high-level language. C allows users to access memory block directly and allows to pass by reference, whereas in Java everything is passed by value and the user doesn't need to manage the memory because this task is performed by the Java garbage collector.

Also, JAVA supports Method Overloading while C does not support overloading at all. JAVA supports function or method overloading-that is we can have two or more functions with the same name(with certain varying parameters like return types to allow the machine to differentiate between them).

Unlike C, JAVA does not support Preprocessors. The preprocessor directives like `#include` & `#define`, etc are considered one of the most essential elements of C programming. However, there are no preprocessors in JAVA. JAVA uses other alternatives for the preprocessors. For instance, `public static final` is used instead of the `#define` preprocessor.

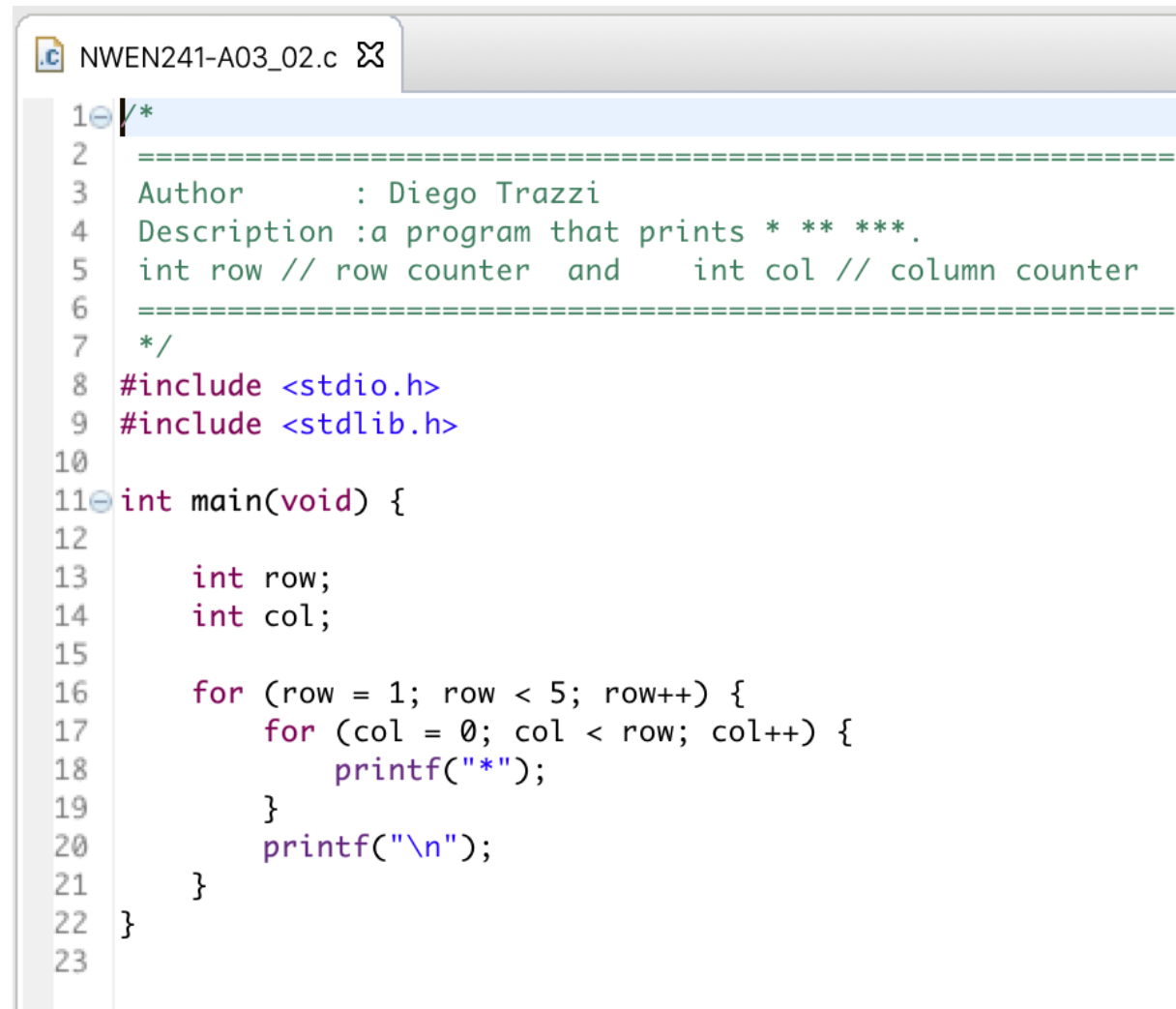
Finally, when an error occurs in a Java program it results in an exception being thrown. It can then be handled using various exception handling techniques. While in C, if there's an error, there IS an error.

Write a program that finds the smallest of several integers entered by user (source code available as attachment):

```
NWEN241A03.c
1  /*
2  =====
3  Author      : Diego Trazzi
4  Description : A program that finds the smallest of several integers entered by user
5  =====
6
7  int number; number of integers to be entered  int value; value input by user
8  int smallest; smallest number
9  int i; counter
10
11  */
12  #include <stdio.h>
13  #include <stdlib.h>
14  #include <limits.h>
15
16  int main(void) {
17
18      int number;
19      int smallest = INT_MAX;
20      int value;
21
22      printf("Number of integers to be compared: ");
23      scanf("%d", &number); // & gets the memory address
24
25      for (int i = 0; i < number; i++) {
26          printf("Please enter a value: ");
27          scanf("%d", &value);
28          if (value < smallest) {
29              smallest = value;
30          }
31      }
32
33      printf("Smallest number you entered is: %d\n", smallest);
34
35  }
36
```

Write a program that prints the following pattern. Use for loops to generate the pattern. All asterisks (*) should be printed by a single printf statement of the form printf("*"). You can only use the following variables.

```
int row; /* row counter */ int col; /* column counter */
```



```
1  /*
2  =====
3  Author      : Diego Trazzi
4  Description : a program that prints * ** ***.
5  int row // row counter and int col // column counter
6  =====
7  */
8  #include <stdio.h>
9  #include <stdlib.h>
10
11 int main(void) {
12     int row;
13     int col;
14
15     for (row = 1; row < 5; row++) {
16         for (col = 0; col < row; col++) {
17             printf("*");
18         }
19         printf("\n");
20     }
21 }
22
23
```

Using the switch statement, write a program that counts and prints the number of each different letter grade user entered. Assume we have only five different grades – “A” / “a”, “B” / “b”, “C” / “c”, “D” / “d”, “E” / “e”. You need to provide a special character to end the input (user needs to enter this character to end the input). You can only use the following variables.

int **grade**; /* current grade. You can also use char grade. You may use either getchar or scanf to handle the input. */

int **aCount** = 0; /* total a grades */

int **bCount** = 0; /* total b grades */

int **cCount** = 0; /* total c grades */

int **dCount** = 0; /* total d grades */

int **eCount** = 0; /* total e grades */

```
*NWEN241-A03_03.c
1 /*
2 =====
3 Author      : Diego Trazzi
4 Description : a program that counts and prints the number of each different letter grade user entered
5 =====
6 */
7
8 #include <stdio.h>
9 #include <stdlib.h>
10 #include <ctype.h>
11
12 int main(void) {
13     char grade; /* current grade. You can also use char grade. You may use either getchar or scanf to handle the input. */
14     int aCount = 0; /* total a grades */
15     int bCount = 0; /* total b grades */
16     int cCount = 0; /* total c grades */
17     int dCount = 0; /* total d grades */
18     int eCount = 0; /* total e grades */
19
20     printf("Enter grade (A-F,a-f). Enter x to terminate.\n");
21
22     while (grade != 'X') {
23         grade = toupper(getchar());
24         switch (grade) {
25             case ('A'):
26                 aCount++;
27                 break;
28             case ('B'):
29                 bCount++;
30                 break;
31             case ('C'):
32                 cCount++;
33                 break;
34             case ('D'):
35                 dCount++;
36                 break;
37             case ('E'):
38                 eCount++;
39                 break;
40         }
41     }
42
43     printf("Total number of A: %d\n", aCount);
44     printf("Total number of B: %d\n", bCount);
45     printf("Total number of C: %d\n", cCount);
46     printf("Total number of D: %d\n", dCount);
47     printf("Total number of E: %d\n", eCount);
48
49     return (0);
50 }
51
52
```