**Lab 03: Introduction the DS4 and Functions**

**LAB # 4**

**SECTION # 4**

**James Gaul**

**Submitted: 2/22/24**

**Lab Date: 2/16/24**

# Problem 1: Compiler Errors

**Lab04-1\_1**

* 1. Errors/Changes Made:
     1. Line 28/29: Added a missing semicolon
     2. Line 36/37: Added missing quotation marks around a string.
     3. Line 43/44: Replaced missing bracket after “else”
     4. Line 46/47: Corrected “pritf” to “printf”
  2. Initial Error:

A screen shot of a computer program

Description automatically generated

* 1. Edited Source Code:

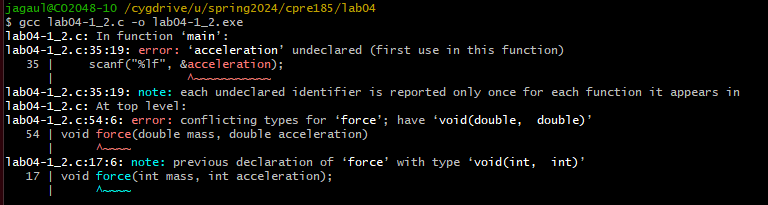
1. /\*----------------------------------------------------------------------------
2. - SE 185: Lab 04 - Debugging Code -
3. - Name: James Gaul -
4. - Section: 3 -
5. - NetID: 947125207 -
6. - Date: 2/16/24 -
7. -----------------------------------------------------------------------------\*/
8. /\*----------------------------------------------------------------------------
9. - Includes -
10. -----------------------------------------------------------------------------\*/
11. #include <stdio.h>
12. /\*----------------------------------------------------------------------------
13. - Notes -
14. -----------------------------------------------------------------------------\*/
15. // Compile with gcc lab04-1\_1.c -o lab04-1\_1
16. // Run with ./lab04-1\_1
17. /\* This program outputs if a integer will divide into another integer with no remainder. \*/
18. /\*----------------------------------------------------------------------------
19. - Implementation -
20. -----------------------------------------------------------------------------\*/
21. int main**(**int argc**,** char **\***argv**[])**
22. **{**
23. int i**,** j**;**
24. //printf("Enter an integer: ")
25. printf**(**"Enter an integer: "**);**
27. scanf**(**"%d"**,** **&**i**);**
28. //printf("Enter another integer: );
29. printf**(**"Enter another integer: "**);**
31. //scanf("%d", &j)
32. scanf**(**"%d"**,** **&**j**);**
33. **if** **(**j **%** i **==** 0**)**
34. **{**
35. printf**(**"%d divides %d\n"**,** i**,** j**);**
36. //} else
37. **}** **else{**
38. //pritf("%d does not divide %d\n", i, j);
39. printf**(**"%d does not divide %d\n"**,** i**,** j**);**
41. printf**(**"%d %% %d is %d\n"**,** j**,** i**,** **(**j **%** i**));**
42. **}**
43. **return** 0**;**
44. **}**
    1. Corrected Output:

**A screen shot of a computer

Description automatically generated**

**Lab04-1\_2**

1. Errors/Changes Made:
   1. Line 18/19: Changed function variables from “int” to “double”
   2. Line 34/35: Added variable declaration of “acceleration” for the function “main”
2. Initial Error:



1. Edited Source Code:
2. /\*----------------------------------------------------------------------------
3. - SE 185: Lab 04 - Debugging Code -
4. - Name: James Gaul -
5. - Section: 3 -
6. - NetID: 947125207 -
7. - Date: 2/16/24 -
8. -----------------------------------------------------------------------------\*/
9. /\*----------------------------------------------------------------------------
10. - Includes -
11. -----------------------------------------------------------------------------\*/
12. #include <stdio.h>
13. /\*----------------------------------------------------------------------------
14. - Prototypes -
15. -----------------------------------------------------------------------------\*/
16. //void force(int mass, int acceleration);
17. void force**(**double mass**,** double acceleration**);**
18. /\*----------------------------------------------------------------------------
19. - Notes -
20. -----------------------------------------------------------------------------\*/
21. // Compile with gcc lab04-1\_2.c -o lab04-1\_2
22. // Run with ./lab04-1\_2
23. /\* This program takes two inputs, acceleration and mass,
24. \* and outputs the force = mass \* acceleration \*/
25. /\*----------------------------------------------------------------------------
26. - Implementation -
27. -----------------------------------------------------------------------------\*/
28. int main**(**int argc**,** char **\***argv**[])**
29. **{**
30. //double mass;
31. double mass**,** acceleration**;**
33. printf**(**"Enter an acceleration in m/s^2: "**);**
34. scanf**(**"%lf"**,** **&**acceleration**);**
35. printf**(**"Enter the mass of the object in kg: "**);**
36. scanf**(**"%lf"**,** **&**mass**);**
37. printf**(**"\nYou entered %lf m/s^2.\n"**,** acceleration**);**
38. printf**(**"You entered %lf kg.\n\n"**,** mass**);**
39. force**(**mass**,** acceleration**);**
40. **return** 0**;**
41. **}**
42. /\*\*
43. \* Given mass and acceleration, calculates the force exerted.
44. \*
45. \* **@param** mass - The given mass of an object in kilograms.
46. \* **@param** acceleration - The acceleration of an object in m/s^2.
47. \*/
48. void force**(**double mass**,** double acceleration**)**
49. **{**
50. printf**(**"The force is approximately %.2lf Newtons.\n"**,** mass **\*** acceleration**);**
51. **}**
52. Fixed Output:

A screen shot of a computer

Description automatically generated

**Lab04-1\_3**

1. Errors/Changes Made:
   * Line 13: Added stdio.h inclusion
   * Line 14: Added stdlib.h inclusion
2. Initial Error:

A screenshot of a computer program

Description automatically generated

1. Reduced errors after first line changed (adding “#include <stdio.h>”)

A screenshot of a computer program

Description automatically generated

1. Edited Code
2. /\*----------------------------------------------------------------------------
3. - SE 185: Lab 04 - Debugging Code -
4. - Name: James Gaul -
5. - Section: 3 -
6. - NetID: 947125207 -
7. - Date: 2/16/24 -
8. -----------------------------------------------------------------------------\*/
9. /\*----------------------------------------------------------------------------
10. - Includes -
11. -----------------------------------------------------------------------------\*/
12. #include <time.h>
13. #include <stdio.h> //Added inclusion
14. #include <stdlib.h> //Added inclusion
15. /\*----------------------------------------------------------------------------
16. - Prototypes -
17. -----------------------------------------------------------------------------\*/
18. void hoo**();**
19. void print\_face**(**int selection**);**
20. /\*----------------------------------------------------------------------------
21. - Notes -
22. -----------------------------------------------------------------------------\*/
23. /\* This is a simple program that takes a user inputs
24. \* and prints out a message based on that input \*/
25. // Compile with gcc lab04-1\_3.c -o lab04-1\_3
26. // Run with ./lab04-1\_3
27. /\*----------------------------------------------------------------------------
28. - Implementation -
29. -----------------------------------------------------------------------------\*/
30. int main**(**int argc**,** char **\***argv**[])**
31. **{**
32. srand**(**time**(NULL));**
33. int selection **=** 0**;**
34. printf**(**"Enter 1 for happy, 2 for sad, 3 for neutral, any other integer for random: "**);**
35. scanf**(**"%d"**,** **&**selection**);**
36. **if** **(**selection **<** 1 **||** selection **>** 3**)**
37. **{**
38. selection **=** rand**()** **%** 4**;**
39. **}**
40. print\_face**(**selection**);**
41. **return** 0**;**
42. **}**
43. /\*\*
44. \* Prints a funny face.
45. \*
46. \* **@param** selection - The inputted value which determines which face to print.
47. \*/
48. void print\_face**(**int selection**)**
49. **{**
50. **if** **(**selection **==** 1**)**
51. **{**
52. printf**(**"Have a nice day! :) \n"**);**
53. **}** **else** **if** **(**selection **==** 2**)**
54. **{**
55. printf**(**":(\n"**);**
56. **}** **else** **if** **(**selection **==** 3**)**
57. **{**
58. printf**(**"Meh :\\ \n"**);**
59. **}** **else**
60. **{**
61. hoo**();**
62. **}**
63. **}**
64. /\*\*
65. \* Prints an owl face.
66. \*/
67. void hoo**()**
68. **{**
69. printf**(**" \*\_\_\_\*\n {O,O}\n/)\_\_\_)\n\_\"\_\_\"\_\n"**);**
70. **}**
71. Fixed Output:

A screenshot of a computer program

Description automatically generated

**Lab04-1\_4**

1. Changes Made/Errors:
   1. Changed invalid variable name “speed\_of\_light!” into “speed\_of\_light” in all locations (variable names can’t include special characters)
   2. Changed invalid variable name “wave-length” into “waveLength” in all locations
   3. Changed invalid variable name “~length\_in\_meters” to “length\_in\_meters”
   4. Changed invalid variable name “plank const” to “plankConst” in all locations (variable names can’t include spaces)
   5. Changed invalid variable name “0energy” to “energy” in all locations (variable names can’t start with a digit)
2. Initial Error:

A screen shot of a computer program

Description automatically generated

1. Edited Code:

/\*----------------------------------------------------------------------------

- SE 185: Lab 04 - Debugging Code -

- Name: James Gaul -

- Section: 3 -

- NetID: 947125207 -

- Date: 2/16/24 -

-----------------------------------------------------------------------------\*/

/\*----------------------------------------------------------------------------

- Includes -

-----------------------------------------------------------------------------\*/

#include <stdio.h>

#include <math.h>

/\*----------------------------------------------------------------------------

- Notes -

-----------------------------------------------------------------------------\*/

// Compile with gcc lab04-1\_4.c -o lab04-1\_4

// Run with ./lab04-1\_4

/\* This program calculates the energy of one photon

\* of user-inputted wave-length of light \*/

/\*----------------------------------------------------------------------------

- Implementation -

-----------------------------------------------------------------------------\*/

int main**(**int argc**,** char **\***argv**[])**

**{**

//double speed\_of\_light!;

double speed\_of\_light**;**

//double wave-length;

double waveLength**;**

//double ~length\_in\_meters;

double length\_in\_meters**;**

//double plank const;

double plankConst**;**

//double 0energy;

double energy**;**

//plank const = 6.62606957 \* pow(10, -34); // Planck's constant

plankConst **=** 6.62606957 **\*** pow**(**10**,** **-**34**);**

//speed\_of\_light! = 2.99792458 \* pow(10, 8); // Constant for the speed of light

speed\_of\_light **=** 2.99792458 **\*** pow**(**10**,** 8**);**

//wave-length = 0;

waveLength **=** 0**;**

//~length\_in\_meters = 0;

length\_in\_meters **=** 0**;**

//0energy = 0;

energy **=** 0**;**

printf**(**"Welcome! This program will give the energy, in Joules,\n"**);**

printf**(**"of 1 photon with a certain wave-length.\n"**);**

printf**(**"Please input a wave-length of light in nano-meters.\n"**);**

printf**(**"Please do not enter a negative, or zero, wave-length.\n"**);**

//scanf("%lf", &wave-length);

scanf**(**"%lf"**,** **&**waveLength**);**

//if (wave-length > 0.0)

**if** **(**waveLength **>** 0.0**)**

**{**

//~length\_in\_meters = wave-length / pow(10, 9); // Converting nano-meters to meters

length\_in\_meters **=** waveLength **/** pow**(**10**,** 9**);**

//0energy = (plank const \* speed\_of\_light!) / ~length\_in\_meters; // Calculating the energy of 1 photon

energy **=** **(**plankConst **\*** speed\_of\_light**);**

/\*printf("A photon with a wave-length of %08.3lf nano-meters, carries "

"\napproximately %030.25lf joules of energy.", wave-length, 0energy);

\*/

printf**(**"A photon with a wave-length of %08.3lf nano-meters, carries "

"\napproximately %030.25lf joules of energy."**,** waveLength**,** energy**);**

**}** **else**

**{**

printf**(**"Sorry, you put in an invalid number."**);**

printf**(**"Please rerun the program and try again."**);**

**}**

**return** 0**;**

**}**

1. Fixed Output:

A screen shot of a computer

Description automatically generated

**Lab04-1\_5**

* 1. Errors/Changes Made:
     1. Line 19: Removed “main” redefinition
     2. Lines 44-47 Commented out alternate “main” function (it was never called and had no apparent purpose)
  2. Initial Error:

A screenshot of a computer

Description automatically generated

* 1. Edited Code:

1. /\*----------------------------------------------------------------------------
2. - SE 185: Lab 04 - Debugging Code -
3. - Name: James Gaul -
4. - Section: 3 -
5. - NetID: 947125207 -
6. - Date: 2/16/24 -
7. -----------------------------------------------------------------------------\*/
8. /\*----------------------------------------------------------------------------
9. - Includes -
10. -----------------------------------------------------------------------------\*/
11. #include <stdio.h>
12. /\*----------------------------------------------------------------------------
13. - Prototypes -
14. -----------------------------------------------------------------------------\*/
15. int sum\_function**(**int number**);**
16. //int main(); -Removed main redefinition
17. /\*----------------------------------------------------------------------------
18. - Notes -
19. -----------------------------------------------------------------------------\*/
20. // Compile with gcc lab04-1\_5.c -o lab04-1\_5
21. // Run with ./lab04-1\_5
22. /\* This program calculates the sum of 1 to x, where x is a user input \*/
23. /\*----------------------------------------------------------------------------
24. - Implementation -
25. -----------------------------------------------------------------------------\*/
26. int main**(**int argc**,** char **\***argv**[])**
27. **{**
28. int input**;**
29. printf**(**"Please input a number from to sum up to: "**);**
30. scanf**(**"%d"**,** **&**input**);**
31. printf**(**"The sum of 1 to %d is %d\n"**,** input**,** sum\_function**(**input**));**
32. **return** 0**;**
33. **}**
34. /\*int main(int argc, char \*argv[])
35. {
36. printf("Sum is 32!\n");
37. }
38. \*/
39. /\*\*
40. \* Calculates the sum of 1 to number of a given number.
41. \*
42. \* **@param** number - The number that determines what the sum will stop adding at.
43. \* **@return** - The sum of 1 to the given number.
44. \*/
45. int sum\_function**(**int number**)**
46. **{**
47. **return** **(**number **\*** **(**number **+** 1**))** **/** 2**;**
48. **}**
    1. Fixed Output:

A black screen with white text

Description automatically generated

# Part 2: Unintended Results (Logic Errors)

**Lab04-2\_1**

1. Changes Made/Errors:
   1. Line 34/35: Changed “==” to “=” (double equal signs is used for comparisons, not variable definition)
   2. Lines 40/41 and 46/47: Changed “=” to “==” (single equal sign is used fo variable definition, not comparisons)
2. Initial Output:

A screen shot of a computer program

Description automatically generated

1. Edited Source Code:
2. /\*----------------------------------------------------------------------------
3. - SE 185: Lab 04 - Debugging Code -
4. - Name: James Gaul -
5. - Section: 3 -
6. - NetID: 947125207 -
7. - Date: 2/16/24 -
8. -----------------------------------------------------------------------------\*/
9. /\*----------------------------------------------------------------------------
10. - Includes -
11. -----------------------------------------------------------------------------\*/
12. #include <stdio.h>
13. /\*----------------------------------------------------------------------------
14. - Prototypes -
15. ----------------------------------------------------------------------------\*/
16. int is\_odd**(**int number**);**
17. int is\_even**(**int number**);**
18. /\*----------------------------------------------------------------------------
19. - Notes -
20. -----------------------------------------------------------------------------\*/
21. // Compile with gcc lab04-2\_1.c -o lab04-2\_1
22. // Run with ./lab04-2\_1
23. /\* This program accepts a user input and determines
24. \* if the integer is an odd or an even number \*/
25. /\*----------------------------------------------------------------------------
26. - Implementation -
27. -----------------------------------------------------------------------------\*/
28. int main**(**int argc**,** char **\***argv**[])**
29. **{**
30. //int input == 0;
31. int input **=** 0**;**
32. printf**(**"Please input an integer: "**);**
33. scanf**(**"%d"**,** **&**input**);**
34. //if (is\_odd(input) = 1)
35. **if** **(**is\_odd**(**input**)** **==** 1**)**
36. **{**
37. printf**(**"%d is an odd number!\n"**,** input**);**
38. **}**
39. //if (is\_even(input) = 1)
40. **if** **(**is\_even**(**input**)** **==** 1**)**
41. **{**
42. printf**(**"%d is an even number!\n"**,** input**);**
43. **}**
44. **return** 0**;**
45. **}**
46. /\*\*
47. \* Determines whether the given number is even.
48. \*
49. \* **@param** number - The number in question of even status.
50. \* **@return** - True if the given number was even.
51. \*/
52. int is\_even**(**int number**)**
53. **{**
54. **return** **!(**number **%** 2**);**
55. **}**
56. /\*\*
57. \* Determines whether the given number is odd.
58. \*
59. \* **@param** number - The number in question of odd status.
60. \* **@return** - True if the given number was odd.
61. \*/
62. int is\_odd**(**int number**)**
63. **{**
64. **return** number **%** 2**;**
65. **}**
66. Fixed Output:

A computer screen with yellow text

Description automatically generated

**Lab04-2\_2**

1. Changes Made/Errors:
   * Changed text of if/else statements, using a simple “greater than or equal to” statements to compare values.
2. Initial Output:

A black screen with white text

Description automatically generated

1. Edited Source Code:

/\*----------------------------------------------------------------------------

- SE 185: Lab 04 - Debugging Code -

- Name: James Gaul -

- Section: 3 -

- NetID: 947125207 -

- Date: 2/16/24 -

-----------------------------------------------------------------------------\*/

/\*----------------------------------------------------------------------------

- Includes -

-----------------------------------------------------------------------------\*/

#include <stdio.h>

/\*----------------------------------------------------------------------------

- Prototypes -

-----------------------------------------------------------------------------\*/

void how\_many\_whole\_digits**(**int number**);**

/\*----------------------------------------------------------------------------

- Notes -

-----------------------------------------------------------------------------\*/

/\* This program calculates the number of digits in a number from 1 to 10000000 \*/

// Compile with gcc lab04-2\_2.c -o lab04-2\_2

// Run with ./lab04-2\_2

/\*----------------------------------------------------------------------------

- Implementation -

-----------------------------------------------------------------------------\*/

int main**(**int argc**,** char **\***argv**[])**

**{**

int input**;**

printf**(**"Please input an integer from 1 up to 10000000: "**);**

scanf**(**"%d"**,** **&**input**);**

**if** **(**input **>** 10000000 **||** input **<** 1**)**

**{**

printf**(**"Invalid number!\n"**);**

**return** **-**1**;**

**}**

how\_many\_whole\_digits**(**input**);**

**return** 0**;**

**}**

/\*\*

\* This function divides a number by the 10^n, to

\* see if the divided number has "n" digits

\*

\* **@param** number - The number to determine how many whole digits exist within.

\*/

void how\_many\_whole\_digits**(**int number**)**

**{**

//if ((double) number / 10000000 != 0)

**if** **((**double**)** number **>=** 10000000**)**

**{**

printf**(**"8 digits\n"**);**

**}** //else if ((double) number / 1000000 != 0)

**else** **if** **((**double**)** number **>=** 1000000**)**

**{**

printf**(**"7 digits\n"**);**

**}** //else if ((double) number / 100000 != 0)

**else** **if** **((**double**)** number **>=** 100000**)**

**{**

printf**(**"6 digits\n"**);**

**}** //else if ((double) number / 10000 != 0)

**else** **if** **((**double**)** number **>=** 10000**)**

**{**

printf**(**"5 digits\n"**);**

**}** //else if ((double) number / 1000 != 0)

**else** **if** **((**double**)** number **>=** 1000**)**

**{**

printf**(**"4 digits\n"**);**

**}** //else if ((double) number / 100 != 0)

**else** **if** **((**double**)** number **>=** 100**)**

**{**

printf**(**"3 digits\n"**);**

**}** //else if ((double) number / 10 != 0)

**else** **if** **((**double**)** number **>=** 10**)**

**{**

printf**(**"2 digits\n"**);**

**}** //else if ((double) number / 1 != 0)

**else** **if** **((**double**)** number **>=** 1**)**

**{**

printf**(**"1 digit\n"**);**

**}**

**}**

1. Fixed Output:

A black screen with white text

Description automatically generated

**Lab04-2\_3**

* 1. Changes Made/Errors:
     1. Line 37/38: In “scanf” statement, changed variable identifiers from “%lf” (for long format variables) to “%d” (for integer variables)
  2. Initial Output:

A screenshot of a computer

Description automatically generated

* 1. Edited Source Code:

1. /\*----------------------------------------------------------------------------
2. - SE 185: Lab 04 - Debugging Code -
3. - Name: James Gaul -
4. - Section: 3 -
5. - NetID: 947125207 -
6. - Date: 2/16/24 -
7. -----------------------------------------------------------------------------\*/
8. /\*----------------------------------------------------------------------------
9. - Includes -
10. -----------------------------------------------------------------------------\*/
11. #include <stdio.h>
12. /\*----------------------------------------------------------------------------
13. - Prototypes -
14. -----------------------------------------------------------------------------\*/
15. void variable\_swap**(**int i**,** int j**);**
16. void math\_swap**(**int i**,** int j**);**
17. /\*----------------------------------------------------------------------------
18. - Notes -
19. -----------------------------------------------------------------------------\*/
20. /\* This program accepts two integers as user input and
21. \* swaps their values using two different methods \*/
22. // Compile with gcc lab04-2\_3.c -o lab04-2\_3
23. // Run with ./lab04-2\_3
24. /\*----------------------------------------------------------------------------
25. - Implementation -
26. -----------------------------------------------------------------------------\*/
27. int main**(**int argc**,** char **\***argv**[])**
28. **{**
29. int first **=** 0**,** second **=** 0**;**
30. printf**(**"Please input two integers separated by a space: "**);**
31. //scanf("%lf %lf", &first, &second);
32. scanf**(**"%d %d"**,** **&**first**,** **&**second**);**
33. printf**(**"\n"**);**
34. variable\_swap**(**first**,** second**);**
35. printf**(**"\n"**);**
36. math\_swap**(**first**,** second**);**
37. **return** 0**;**
38. **}**
39. /\*\*
40. \* Swaps the values of two integers using a temp variable.
41. \*
42. \* **@param** i - The first value to be swapped.
43. \* **@param** j - The second value to be swapped.
44. \*/
45. void variable\_swap**(**int i**,** int j**)**
46. **{**
47. printf**(**"Now doing a swap using an extra variable: \n"**);**
48. printf**(**"Before Swap: First: %d, Second: %d\n"**,** i**,** j**);**
49. int temp **=** i**;**
50. i **=** j**;**
51. j **=** temp**;**
52. printf**(**"After Swap: First: %d, Second: %d\n"**,** i**,** j**);**
53. **}**
54. /\*\*
55. \* Swaps the values of two integers without using a temp variable.
56. \*
57. \* **@param** i - The first value to be swapped.
58. \* **@param** j - The second value to be swapped.
59. \*/
60. void math\_swap**(**int i**,** int j**)**
61. **{**
62. printf**(**"Now doing a swap using addition and subtraction: \n"**);**
63. printf**(**"Before Swap: First: %d, Second: %d\n"**,** i**,** j**);**
64. i **=** i **+** j**;**
65. j **=** i **-** j**;**
66. i **=** i **-** j**;**
67. printf**(**"After Swap: First: %d, Second: %d\n"**,** i**,** j**);**
68. **}**
    1. Fixed Output:

A screenshot of a computer program

Description automatically generated

**Lab04-2\_4**

1. Changes Made/Errors:
   * 1. Line 37/38: Changed “int” variable declarations to “double”, consistent with the rest of the program.
2. Initial Output:

A screen shot of a computer

Description automatically generated

1. Edited Source Code:

/\*----------------------------------------------------------------------------

- SE 185: Lab 04 - Debugging Code -

- Name: James Gaul -

- Section: 3 -

- NetID: 947125207 -

- Date: 2/16/24 -

-----------------------------------------------------------------------------\*/

/\*----------------------------------------------------------------------------

- Includes -

-----------------------------------------------------------------------------\*/

#include <stdio.h>

/\*----------------------------------------------------------------------------

- Prototypes -

-----------------------------------------------------------------------------\*/

double voltage**(**double resistance**,** double current**);**

double resistance**(**double voltage**,** double current**);**

double current**(**double voltage**,** double resistance**);**

/\*----------------------------------------------------------------------------

- Notes -

-----------------------------------------------------------------------------\*/

// Compile with gcc lab04-2\_4.c -o lab04-2\_4

// Run with ./lab04-2\_4

/\* This program calculates values of resistances,

\* voltages, or current using Ohm's Law \*/

/\*----------------------------------------------------------------------------

- Implementation -

-----------------------------------------------------------------------------\*/

int main**(**int argc**,** char **\***argv**[])**

**{**

int selection **=** 0**;**

//int v, i, r;

double v**,** i**,** r**;**

printf**(**"selection:\n1 for voltage\n2 for resistance\n3 for current\n"**);**

scanf**(**"%d"**,** **&**selection**);**

**if** **(**selection **>** 3 **||** selection **<** 1**)**

**{**

printf**(**"Invalid number\n"**);**

**return** **-**1**;**

**}**

printf**(**"Enter floating point numbers for input...\n"**);**

**if** **(**selection **==** 1**)**

**{**

printf**(**"Please enter a resistance value: "**);**

scanf**(**"%lf"**,** **&**r**);**

printf**(**"Please enter a current value: "**);**

scanf**(**"%lf"**,** **&**i**);**

printf**(**"Your voltage is: %lf Volts\n"**,** voltage**(**r**,** i**));**

**}** **else** **if** **(**selection **==** 2**)**

**{**

printf**(**"Please enter a voltage value: "**);**

scanf**(**"%lf"**,** **&**v**);**

printf**(**"Please enter a current value: "**);**

scanf**(**"%lf"**,** **&**i**);**

printf**(**"Your Resistance is: %lf Ohms\n"**,** resistance**(**v**,** i**));**

**}** **else** **if** **(**selection **==** 3**)**

**{**

printf**(**"Please enter a resistance value: "**);**

scanf**(**"%lf"**,** **&**r**);**

printf**(**"Please enter a voltage value: "**);**

scanf**(**"%lf"**,** **&**v**);**

printf**(**"Your current is: %lf Amps\n"**,** current**(**v**,** r**));**

**}**

**return** 0**;**

**}**

/\*\*

\* Given the resistance and current, calculates and returns the voltage.

\*

\* **@param** resistance - The resistance used to calculate the voltage.

\* **@param** current - The current used to calculate the voltage.

\* **@return** - The voltage calculated from the resistance and current.

\*/

double voltage**(**double resistance**,** double current**)**

**{**

**return** resistance **\*** current**;**

**}**

/\*\*

\* Given the voltage and current, calculates and returns the resistance.

\*

\* **@param** voltage - The voltage used to calculate the resistance.

\* **@param** current - The resistance used to calculate the resistance.

\* **@return** - The resistance calculated from the voltage and current.

\*/

double resistance**(**double voltage**,** double current**)**

**{**

**return** voltage **/** current**;**

**}**

/\*\*

\* Given the voltage and resistance, calculates and returns the current.

\*

\* **@param** voltage - The voltage used to calculate the current.

\* **@param** resistance - The resistance used to calculate the current.

\* **@return** - The current calculated from the voltage and resistance.

\*/

double current**(**double voltage**,** double resistance**)**

**{**

**return** voltage **/** resistance**;**

**}**

1. Fixed Output:

A screenshot of a computer program

Description automatically generated

**Lab04-2\_5**

1. Changes Made/Errors:
   * 1. Line 47/48: Changed “%lf” to “%d”, to match an integer variable.
     2. Line 114/115: Changed variable name from “n” to “number”.
2. Initial Output:

A screenshot of a computer

Description automatically generated

1. Fixed Code:

A black background with white text

Description automatically generated

Problem 3: Putting it all together

**Lab04-3\_1**

1. Changes Made/Errors
   1. Line 13: added inclusion of “stdlib.h”, so that the “srand” function can be used.
   2. Line 16: Fixed missing “/\*” to multiline comment, which caused inclusions to be ignored.
   3. Line 24: Added missing declaration of function “run\_game”
   4. Line 36: Fixed missing “\*/”, which commented out variable declarations of the “main()” function.
   5. Line 53/54: fixed a typo in the variable name, changing “playd” to “played”
   6. Line 76/77: Replaced a missing “&” in the scanf statement
   7. Line 107: Added variable definition.
   8. Line 111/112: Changed “%c” to “%d”, as the input variable was an int rather than a char.
   9. Line 120/121: changed “=” to “==”, as the if statement seeks to compare values, not assign them.
2. Initial Output:

A computer screen with text on it

Description automatically generated

1. Output after fixing compiler errors:
   1. An error in a scanf statement (using %c rather than %d) results in a segmentation fault.

A screen shot of a computer

Description automatically generated

1. Output after fixing Segmentation fault error
   * The code automatically proceeds without input, as the & statement is missing from the scanf line.

A screenshot of a computer

Description automatically generated

1. Output after fixing scanf() error
   * An “if” statement included “number = computer\_number” instead of “number == computer\_number”. This automatically set the number to the requested value, resulting in automatically returning “correct”.

A screenshot of a computer

Description automatically generated

1. Final Edited Source Code:
2. /\*----------------------------------------------------------------------------
3. - SE 185: Lab 04 - Debugging Code -
4. - Name: James Gaul -
5. - Section: 3 -
6. - NetID: 947125207 -
7. - Date: 2/16/24 -
8. -----------------------------------------------------------------------------\*/
9. /\*----------------------------------------------------------------------------
10. - Includes -
11. -----------------------------------------------------------------------------\*/
12. #include <stdio.h>
13. #include <stdlib.h> //Added inclusion
14. #include <time.h>
15. //fixed comment error on prototypes
16. /\*-----------------------------------------------------------------------------
17. - Prototypes -
18. ------------------------------------------------------------------------------\*/
19. char ask\_to\_play**(**int times\_played**);**
20. int select\_random\_number**();**
21. //declared function run\_game
22. void run\_game**(**int computer\_number**);**
23. /\*----------------------------------------------------------------------------
24. - Notes -
25. -----------------------------------------------------------------------------\*/
26. // Compile with gcc lab04-3.c -o lab04-3
27. // Run with ./lab04-3
28. /\* This program will play a simple Guessing Game with the computer. \*/
29. /\*----------------------------------------------------------------------------
30. - Implementation -
31. -----------------------------------------------------------------------------\*/
32. //fixed implementation header commenting
33. int main**(**int argc**,** char **\***argv**[])**
34. **{**
35. char prompt **=** '-'**;**
36. int played **=** 0**;**
37. int computer\_guess **=** 0**;**
38. prompt **=** ask\_to\_play**(**played**);**
39. played **=** 1**;**
41. **while** **(**prompt **==** 'y'**)** /\* This line does not contain an error \*/
42. **{**
43. computer\_guess **=** select\_random\_number**();**
44. run\_game**(**computer\_guess**);**
45. //prompt = ask\_to\_play(playd);
46. prompt **=** ask\_to\_play**(**played**);**
47. **}**
48. printf**(**"\n\nThanks for playing!\n"**);**
49. **return** 0**;**
50. **}**
51. /\*\*
52. \* Asks the player if they want to play the Guessing Game.
53. \*
54. \* **@param** played\_before - Whether the player has played a round of the game before or not.
55. \* **@return** - Whether the player wants to play again or not.
56. \*/
57. char ask\_to\_play**(**int played\_before**)**
58. **{**
59. char yes\_or\_no**;**
60. **if** **(!**played\_before**)** /\* This line does not contain an error \*/
61. **{**
62. printf**(**"Do you want to play a game? "
63. "Enter 'y' to play, anything else not to play. :(\n -> "**);**
64. //scanf(" %c", yes\_or\_no);
65. scanf**(**" %c"**,** **&**yes\_or\_no**);**
66. **}** **else**
67. **{**
68. scanf**(**" %c"**,** **&**yes\_or\_no**);**
69. **}**
70. printf**(**"%c"**,** yes\_or\_no**);**
71. **return** yes\_or\_no**;**
72. **}**
73. /\*\*
74. \* Generates a random number between 1 to 100, inclusive.
75. \*
76. \* **@return** - A number between 1 and 100, inclusive.
77. \*/
78. int select\_random\_number**()**
79. **{**
80. srand**(**time**(NULL));**
81. **return** rand**()** **%** 100**;**
82. **}**
83. /\*\*
84. \* Starts the Guessing Game for you to play!
85. \*
86. \* **@param** computer\_number - The randomly generated number to be used for the game.
87. \*/
88. void run\_game**(**int computer\_number**)**
89. **{**
90. int number **=** 0**;**
91. int correct **=** 0**;** //added variable declaration
92. printf**(**"\n\nYou are guessing a number. The options are 1 through 100.\n\n"**);**
93. printf**(**"What is your guess on what number I will select?\n -> "**);**
94. //scanf("%c", &number); Source of core dump error
95. scanf**(**"%d"**,** **&**number**);**
96. **while** **(!**correct**)** /\* This line does not contain an error \*/
97. **{**
98. **if** **(**number **<** 1 **||** number **>** 100**)**
99. **{**
100. printf**(**"\nYour number is not within the correct range of numbers. Guess again\n -> "**);**
101. **}**
102. //else if (number = computer\_number)
103. **else** **if** **(**number **==** computer\_number**)**
104. **{**
105. printf**(**"\nThe number was %d!\n"**,** computer\_number**);**
106. printf**(**"\nYou guessed the number correctly!\n\n"
107. "Do you want to play again? ('y' for yes)\n -> "**);**
108. correct **=** 1**;**
109. **}** //else if (number < computer\_number);
110. **else** **if** **(**number **<** computer\_number**)**
111. **{**
112. printf**(**"\nYou guessed too low. Enter another guess.\n -> "**);**
113. **}** **else**
114. **{**
115. printf**(**"\n You guessed too high. Enter another guess.\n -> "**);**
116. **}**
117. scanf**(**"%d"**,** **&**number**);**
118. **}**
119. **}**
120. Final Output:

A screenshot of a computer

Description automatically generated