**Challenge 1 – Norm Gershon**

**Part 1 Questions**

1. Program would need instructions on how to move and specific directions and distances to move in a specific order.
2. Source code is text language that is used by people to provide instructions to a variety of CPUs. Source code is converted into the elementary CPU-specific instructions known as machine code. The CPU executes these instructions. Instructions /data stored in hard drive then moved to RAM for execution.
3. IDE’s provide all the tools you need to write/run programs in specific languages.
4. Objective-C is a compiled language, ready to run/distribute for specific devices and often providing faster performance. Apps using an interpreted language might run slower and code couldn’t be kept proprietary.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Part 2 Book Challenges**

**Chapter 3**

**Two Floats**

#include <stdio.h>

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

{

float var1 = 3.14;

float var2 = 42.0;

double var3 = var1 + var2;

printf("The sum of the variables is %f.\n", var3);

return 0;

}

=========================================

**Chapter 4**

**The first expression is true.**

=========================================

**Chapter 5**

**Triangle**

float remainingAngle(a, c)

{

return 180.0 - a - c;

}

=========================================

**Chapter 6**

#include <stdio.h>

#include <math.h>

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

{

printf("The sine of 1 radian is %.3f\n", sin(1));

return 0;

}

========================================

**Chapter 7**

#include <stdio.h>

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

{

int i;

for (i = 99; i >= 0; i -= 3) {

printf("%d\n", i);

if (i % 5 == 0) {

printf("Found one!\n");

}

}

return 0;

}

========================================

**Chapter 8**

#include <stdio.h>

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

{

printf("A float is %zu bytes\n", sizeof(float));

return 0;

}

//The smallest number a short can store is -32,768. The largest is 32,767.

//The largest number an unsigned short can store is 65,535.

========================================

**Chapter 10**

#include <stdio.h>

#include <time.h>

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

{

long secondsSince1970 = time(NULL);

long addTime = secondsSince1970 + 4000000;

struct tm now;

struct tm then;

localtime\_r(&secondsSince1970, &now);

localtime\_r(&addTime, &then);

printf("The time is %d:%d:%d\n", now.tm\_hour, now.tm\_min, now.tm\_sec);

printf("The date in 4 million seconds will be %d-%d-%d\n", then.tm\_mon + 1, then.tm\_mday, then.tm\_year + 1900);

return 0;

}