2.

void Funk(int); // we must declare the function first before using it in the main

int main(){

Funk(49);

}

void Funk(int f) {

f = f \* 2;

}

3.

int roundToNearest(double x) {

return x +.5;

}

4.

#include<iostream>

using namespace std;

int gcd(int, int);

int main() {

int z = gcd(259, 111);

cout << "z: " << z << endl; // output : 37

}

int gcd(int x, int y) {

int d;

if (y == 0) {

d = x;

}

else if (y > 0) {

int remainder = x % y;

d = gcd(y, remainder);

}

return d;

}

gcd(259,111) is 37.

5.

Short local 3

Bool local2[0]

Bool local2[1]

Bool local2[2]

Bool local2[3]

Long long local 1

6.

a) The compiler looks for iostream in the system library because it is in angled brackets.

b)The compiler looks for mymath.h header file in the current directory containing the source file and paths specified as part of compiler/IDE settings.

c) The compiler cannot locate the header file because it is not located in the directory containing the source file or paths specified to be part of the compiler.

d)The moral of the story is to keep the header file in the directory containing the source file to be safe of problems.

7.

a) The inline keyword makes it so the compiler inserts the code the function instead of calling it to avoid the function call overhead.

b)Function overloading is when we can make many functions with the same name but with different parameters.

#include<iostream>

#include<random>

#include<cmath>

using namespace std;

int main() {

int seed, angle, gunpowder, speed;

bool run = true;

cout << "Please enter a positive integer seed value (under 1000): " << endl;

cin >> seed;

while (seed >= 1000) {

cin >> seed;

}

default\_random\_engine engine(seed);

uniform\_real\_distribution<double> distr(seed, 1000);

double random = distr(engine), temp = random, sine, cosine, sine2, distance, hit;

printf("The target is %.1f away\n", temp);

while (run == true) {

cout << "Please enter an angle within 0 and 90 degrees: " << endl;

cin >> angle;

cout << "Please enter an amount of gunpowder in kilograms: " << endl;

cin >> gunpowder;

speed = gunpowder \* 30;

sine = sin(angle\*3.14159 / 180);

sine2 = sin(2 \* (angle\*3.14159 / 180));

cosine = cos(angle\*3.14159 / 180);

distance = (speed\*speed\*sine2) / 9.81;

cout << distance << endl;

hit = temp - distance;

if (hit > 1 || hit < -1) {

printf("You were %.2f meters off! Try again!\n", hit);

}

else {

cout << "It's a hit!" << endl;

cin >> distance;

run = false;

}

}

}