**Practicing Objects**

We will be using a Die object to practice coding in an Object Oriented Style. The Die object simulates a die value to use in a dice game. Create a project using the code at the bottom of the worksheet. There will be two files, main.cpp and Die.h

1. In the Die.h file, locate the private variables. What do you think are the purpose of each of these variables?

Int value is the value of each side dice

Int sides is the number of sides of the dice

1. Locate the public section of the class? Are there methods there or just method declarations?

These appear to be just method declarations for int roll and int display

1. Look at the roll() method implementation. What variable is being modified? Describe how it is changing.

Int value is being changed. value = rand() % numSides + 1; value will hold a random integer based on number of sides. The calculation makes it so the probability is even for the chance at rolling each side.

1. Notice the setNumberSides variable. What variable is being modified? Describe how it is changing.

setNumberSides gets assigned to the sides public variable. This changes the sides that the die has to the side parameter

1. What is the display() method doing?

Displaying the roll that just took place. The | and -’s make it appear in the terminal as actual die

1. Now look at the constructor. The constructor sets up any defaults in the class. There is one parameter for the constructor, what is it?

Number of sides on the die

1. Go to main.cpp. Notice the Declaration of the Die. What actual parameter is being sent.

6 (sides)

1. Run the program a few times and view the Die, what is the correlation of the actual parameter of the Die and the value of the Die?

The maximum value that the value variable can be.

1. Change the actual parameter of the die to 20. Then run the program a few times and view the results. How does the Die change?

It changes it to a 20 side die

1. Look back at the Constructor in the Die.h class. What is the name of the parameter?

sides

1. In the same constructor, notice the section after the Colon: This is an inline way of declaring variables. It is setting the numSides variable equal to the variable sides.

Is this a question? I am not sure how to answer. I acknowledge this though!

1. Look at the setNumberSides method. What is it assigning to the numSides variable?

The parameter of sides set in the main.cpp file

1. Go back to the main.cpp file. Add a second die with the same number of sides as d1. Call the roll and display methods for the second Die.
2. Run the program. Do the two dice get the same value? Why do you think that the values are different if there is only one variable *value*?

They are two separate instances in the class. Int value is private and doesn’t effect the other instances

1. Go back to the Die.h file. We are going to create a method to be able to use the die value in main(). Create an additional Method declaration in the class by keying in the line…

int getDieValue();

1. Now implement the method below the class. Make sure to correlate it with the class by keying the Scope Resolution Operator (Die::) between the return type and the method name.
2. The only thing the method will do is return the variable “value”. Key in the code.
3. Look back in the main() method. Key in an If statement to compare the value of d1 and the second die. If the two values are the same, print “Doubles”.
4. Test your code to make sure it work. Paste your code back into the worksheet and submit.

**Die.h**

#pragma once

#include <iostream>

using namespace std;

class Die{

public:

Die(int sides); //Constructor

void roll(); //Method Declaration

void display(); //Method Declaration

void setNumberSides(int sides);//setter Declaration

private:

int value;

int numSides;

};//End Class

//Constructor, sets up the class

Die::Die(int sides): numSides(sides){

value = 1;

}

//Method to roll the die

void Die::roll(){

value = rand() % numSides + 1;

}

//Sets the number of sides of the die

void Die::setNumberSides(int sides){

numSides = sides;

}

//Displays die

void Die::display(){

cout << "-----" << endl;

cout << "| " << value << " |" << endl;

cout << "-----" << endl;

}

**Main.cpp**

#include <iostream>

#include <time.h>

#include "Die.h"

using namespace std;

int main() {

srand(time(NULL)); //Set the random seed

//Declare die

Die d1(6);

//Call methods

d1.roll();

d1.display();

}