Problem Set 1: Report

Task 1

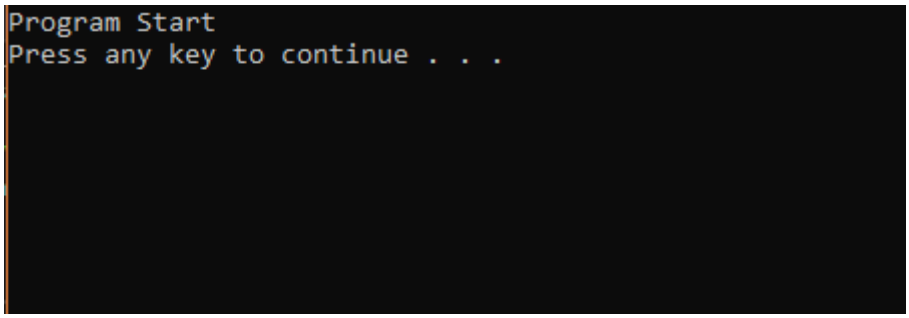

Figure 1: Console Output for Task 1

The program initiates the main loop. The console prints out a “Program Start” message then the system pauses the program, awaiting input before ending the program.

Task 2

A class called Unit was created using header and CPP file. It has a default constructor, an overloaded constructor, 12 attributes, 11 getters and setters, and a deconstructor. As there were no implementations in the main file, no screenshots were taken.

Task 3

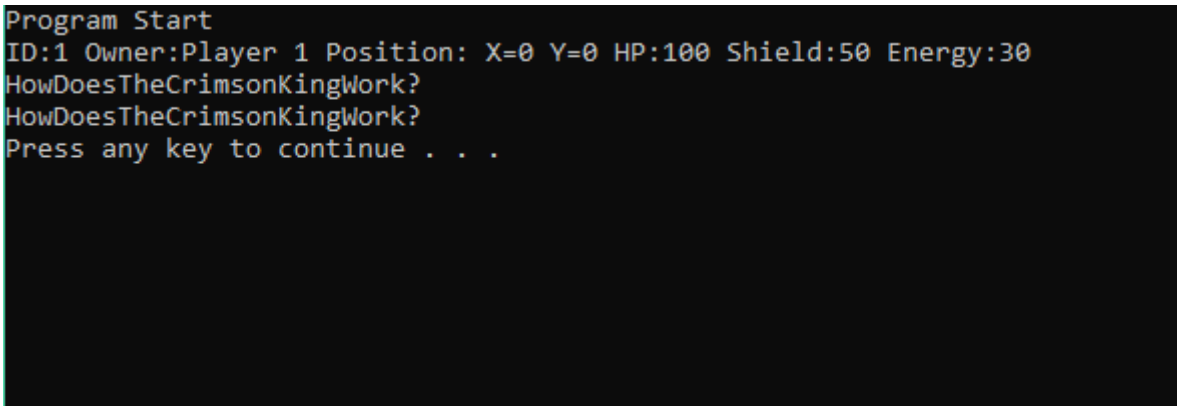

Figure 2: Console Output for Task 3

The console prints out the status of the Unit object via its “PingStatus” function. Then, the insertion operator accepts a message input from the console to be stored in a variable called “fMessage”. After that, the extraction operator outputs the message into the console and the program ends.

Task 4

```
Program Start

Thor 1 Status:
ID:3 Owner:Player 1 Position: X=5 Y=1 HP:250 Shield:0 Energy:75
Thor initiated its attack *Pew Pew*

Viking 1 Status:
ID:2 Owner:Player 1 Position: X=1 Y=1 HP:100 Shield:0 Energy:50
Press any key to continue . . .
```

Figure 3: Console Output for Task 4

Two derived classes of Thor and Viking were added. Both possess attributes and functions from the main class which was the Unit class. Therefore, both of them can use the “PingStatus” function even thou that function was not written inside their header or CPP files. A function called “iniThorAtk” that prints a message is added inside the Thor class to demonstrate the purpose of inheritance.

Task 5

```
Program Start!

Unit's current status:
ID:1 Owner:Player 1 Position: X=1 Y=1 HP:100 Shield:50 Energy:50

Select number to do perfrom actions on the unit
1. Heal unit    2. Damage unit  3. Move unit
Input:3
Movement in X-axis:50
Movement in Y-axis:60

Unit's current status:
ID:1 Owner:Player 1 Position: X=51 Y=61 HP:100 Shield:50 Energy:50

Select number to do perfrom actions on the unit
1. Heal unit    2. Damage unit  3. Move unit
Input:2
Damage amount to unit:60

The unit's shield has been broken! OH NO!

Unit's current status:
ID:1 Owner:Player 1 Position: X=51 Y=61 HP:90 Shield:0 Energy:50

Select number to do perfrom actions on the unit
1. Heal unit    2. Damage unit  3. Move unit
Input:1
Heal amount to unit:40

The unit's HP is at maximum capacity!

Unit's current status:
ID:1 Owner:Player 1 Position: X=51 Y=61 HP:100 Shield:0 Energy:50

Select number to do perfrom actions on the unit
1. Heal unit    2. Damage unit  3. Move unit
Input:2
Damage amount to unit:100

The unit has ceased to exist. This program will now terminate.
Press any key to continue . . .
```

Figure 4: Console Output for Task 5

The program first displays the status of the Unit object via its function. Then, it requests input from the user of actions to be performed on the object. In this case, the movement command was selected and the unit was moved 50 units right in the x-axis and 60 units up in the y-axis. Then, user input was requested yet again. This time the damage option was selected and 60 “damage” was inflicted on the unit. This damage causes the unit’s “Shield” to break so the

units “Shield” is now 0. Additional damage is inflicted on the unit that reduces the units HP by 10. Next, the heal option was selected by the user of an amount of 40. Since the units maximum HP is 100, the current HP only increases by 10. Finally, the damage option is selected again of an amount of 100, which reduces the units HP to 0. This means the unit has “died” thus ending the program.