

Jair Josafath

Assignment 2

1

b, c, e, f, h, and i.

I manually checked using Clion. Integer values and Integer pointers are not the same. Dereferencing (*), which is used to get the value of the data a pointer is pointing at returns in this case the integer value. When "&" is used in front of any variable, the address of the data will be returned. These two symbols should be used with care and behave based on what type of object they are placed in front of. When used incorrectly it will result in compilation errors.

2

- a. Variable m and variable k
- b.
 - 1. No, since n cannot be accessed in B in the first place, because n is a private member of the parent class A.
 - 2. m automatically becomes a private member in B, since m is a protected member in the parent class. Protected members are only accessible by child classes.
 - 3. k will only become private if the private keyword is used where B inherits A.
- c. absolutely.

3

```
#include "Point3D.h"
```

```
Void Point3D:: setPointXZ(uint32_t x, uint32_t z){
```

```
    //since z and x are accessible by Point3D we can use this to access the member.
```

```
    this->z=z;
```

```
    this->x=x;
```

```
    //since there is only one y value, we don't need to use the this keyword
```

```
    y=0;
```

```
}
```

4

Files in the attachments

DynamicArray.cpp and main.cpp

It was hard building the class using .h and .cpp, so I just did it using .cpp