

mainStaticCast.cpp

When running mainStaticCast.cpp with some modifications on variables for “s” and “f” by simply doing,

```
s = static_cast<Student*>(f);  
f = static_cast<Faculty*>(s);
```

It is giving out a compile error saying that there has been a redefinition of variables “s” and “f”. The reason for this is happening is that each class for the variable “s” and “f” are not related by inheritance, meaning class standing for the variable “s” and the class standing for the variable “f” does not match. Also, each class does point to another class. For example, both Student class and Faculty class does not point to each other, but instead they point to the class called Person.

mainDynamicCast.cpp

When running mainDynamicCast.cpp with some modification on variables for “s” and “f” by simply editing,

```
s = dynamic_cast<Student*>(f);  
f = dynamic_cast<Faculty*>(s);
```

it is giving out the runtime error but has compiled correctly because whenever there is a dynamic_cast on each variable, the very first thing that the inherited class reads is the function itself. For example, when s = dynamic_cast<Student*>(f); was called, the faculty pointer is trying to convert to student pointer, and when having a look at the Student class, from the public class, the very first thing that it reads is Student(int, int, int, std::string). In order for dynamic_cast to function properly, the pointer class that has been inherited should have virtuals only. However, in this case, it reads Student(int, int, int, std::string), so that it does compile but has runtime error.