Problem\_Set\_002 – OOP Exercises – Ayoub Ahmad

Problem\_002\_1

1.2 – The Person constructor is called twice because there are two derived classes Employee and Student that are inherited from Person. To avoid this issue in question 1.3, I would use virtual inheritance to make sure that the Person constructor is only called once.

Problem\_002\_2

2.2 – The print function in the Car class is not virtual. Since we are calling a function through the base pointer that is pointing to an object of a derived class, the compiler will use static binding and points to Car, not Toyota or Honda as desired. In order to fix this issue for question 2.3, I would make the function for the Car class virtual and override the derived class functions so the compiler overrides the base function.

Problem\_002\_3

3 – The compile time error is due to the Tiger class being abstract. I have fixed up the code to my liking by:

* Adding public to the classes so that functions can be accessible and overridden by derived classes
* Implemented an override makeSound() function in the derived class
* Call the newly created makeSound() function in main to output “Tiger roars.”, making sure that the code works correctly

4 – Statement 1 is valid because it is safe to point to an object of a derived class by using a pointer to the base class. Statement 2 is not valid because in doing so may allow access to members of Truck which are not present in Vehicle. To fix this issue, we can downcast by using a static cast to directly convert the vehiclePtr to the Truck\*. This is an explicit convert, so only used when you are certain about which objects and types you are working with.