Name: Yunika Upadhayaya

Student ID: 1001631183

11/01/2019

**Answers:**

1. **True**, Edelweiss is a derived class from flower because the function *acquire\_flower* is to return a flower pointer and we are able to return an Edelweiss pointer as well.

2. **False**, just seeing the code fragments alone we cannot assume that the Vase class has a vector of Flower objects and also vase class has the vector of shared pointer of the flower object.

3. **False**, there are only 2 examples of polymorphism in the code: *virtual bool pick\_flower()=0;* and *void operator << (int n).*

4. **False**, because Rose constructor has two parameters: string and int. It cannot be called by passing the integer alone. There should at least be a string parameter.

5. **False,** we cannot change the parameter because we have to delete the pointer as well if we use Flower \*f instead of *shared\_ptr<Flower>f*.

6. **False**, p1<<3 would not be the valid line of code because there is no overloaded operator function in person class.

7. **False,**  the overloaded binary operator defined in the Rose class takes Rose object as one of its operand. We can’t use Rose pointer in place of rose object to use the operator. So, it is not a valid line of code.

8. **True,**  because there is encapsulation (class *Rose*), inheritance (class *Rose* is inherited from class *Flower*) and polymorphism (*virtual bool pick\_flower()=0*).

9. **True,** because Rose class is using the constructor from flower class by passing its string parameter.

10. **False,** we only need to delete 2 pointers as shared pointer free themselves.

11. **True,** because it will only be accessible inside sunflower class.

12. **False,** it will not be possible in the main because in order to inherit from the Flower class, we need to define those functions.

13. **False,**  “<<” is a binary operator whereas “!” is a unary operator. So, the program would not work if we replace it.

14. **False,** we need to use typedef to do so. If we use enum, we won’t be able to change the member variable int *num\_throns* to *Flowerstuff num\_throns.*

15. **False,** because a Message object has not been initialized, but it is called anyway, so we can assume that it is a static function.