1) In your design, you’ve decided to implement classes that have a composition relationship. Give one way of enforcing the major rule of the composition relationship.

-A ‘part’ is controlled by the ‘whole’

2) What is the signature of a function or method?

-The inputs, outputs and function/method name for a function/method.

3) What is overloading?

-Using several functions/methods with the same name that have have different signatures.

4) What is overriding?

-Redefining previously defined functions/methods in subclasses

5) What is an abstract class? Give one reason why we may want to create such a class.

-A class that cannot be instantiated. When objects describe abstract concepts an abstract class ensures that no one can implement it.

6) What effect does a constructor have on an object?

-Initializes new objects

7) What are two differences between instance fields and class fields?

8) What provides the behaviour of an object?

9) What are accessor methods? Why would we use them?

-An Accessor method returns the values of private data members.

10) What are the differences between private, public and protected attributes?

11) What is the interface of a class?

-An interface defines a set of methods and properties that can be inherited by classes.

12) What constitutes an object’s state?

13) What is one technique to enforce a “one-to-one” association relationship between two classes?

14) Can references of a subclass type be used to manipulate objects of its superclass type?

15) What is a copy constructor? Give 2 reasons to incorporate a copy constructor in your design

16) Why/when would you use a protected constructor?

17) Why/when would you use a private constructor?

18) Describe two techniques for improving the robustness of your design, i.e. programming defensively.

19) Many “utility” classes only have static members. On what would you base a decision to have only static members in a class?

20) If you have a class with only static members, it does not need to be instantiated. Give one way of making sure a class cannot be instantiated

21) In your design, you want to control how and when objects are instantiated. Give an example of one strategy you could adopt to include such a feature in your design

22) In an inheritance hierarchy, there are static methods defined in the superclass. Can these be overridden in subclasses?

23) Which UML diagram would you choose to communicate how an object’s attributes’ values changed throughout its lifetime?

24) Which UML diagram would you choose to communicate the structure of the objects in your design