EECS1720
Worksheet 3 – Inheritance

1) How do you determine whether a class extends another given its API? What if you were given the UML class diagram?

2) Given that a class extends another, what can you say about the features of the two classes?

3) If a class extends another, why is it thought of as a specialization?

4) Argue that an inheritance hierarchy can be thought of as several inheritance chains with a class in common.

5) Is it possible for a class to have two children? How about two parents?

6) What is “single inheritance”?

7) What is “multiple inheritance”?

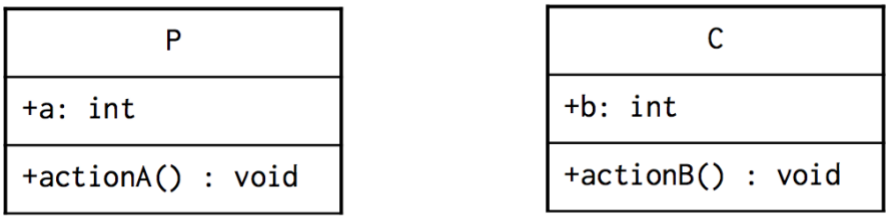
8) If a subclass has a method with the same name as an existing parent’s method, which method will appear in the subclass API? Will your answer change if the child’s method has a different parameter list than that of the parent?

9) If a subclass has a field with the same name as a parent’s field, which of the two will appear in the subclass API? Will your answer change if the child’s field has the same type as that of the parent?

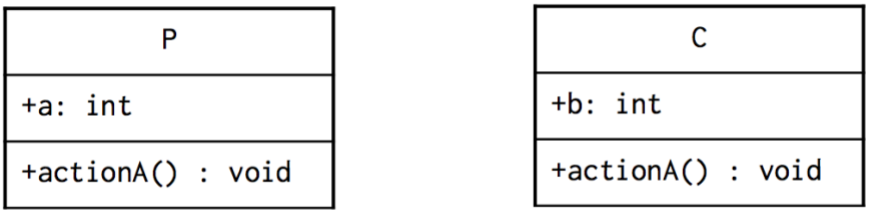
10) Given UML class diagrams for C and P, determine fields and methods that would be found in C's API (like the following figure)

Lets assume C extends P, show the API for C given each of the following =>

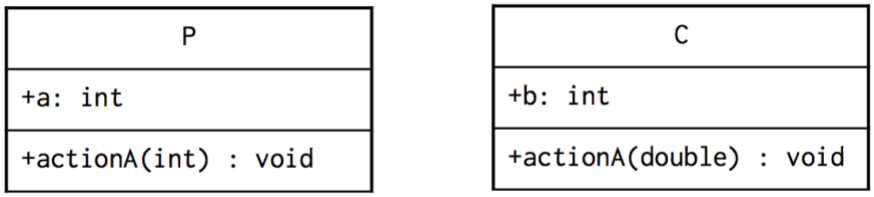
(i)



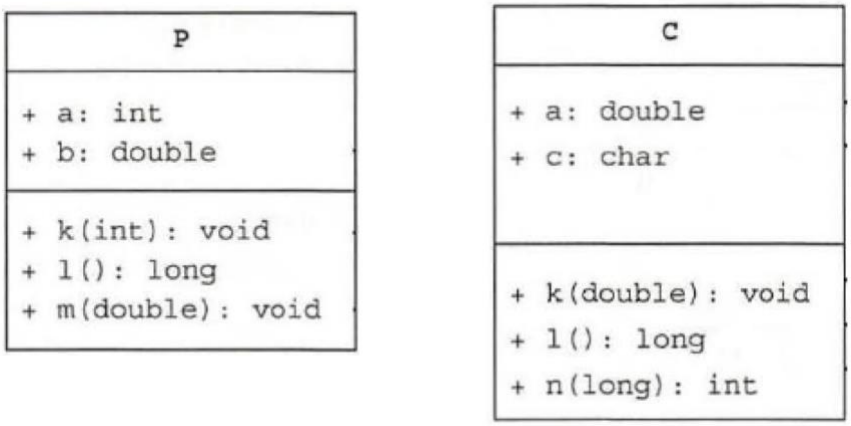
(ii)



(iii)



(iv)



11) What is an interface?

12) How are the following "features" represented in UML?
    a. Aggregation
    b. Composition
    c. A standalone class
    d. A class with private methods and public attributes?
    e. A static field?
    f. A static method?
    g. An interface?
    h. Private vs. public vs. protected fields / methods?
    i. Inheritance
    j. Interfaces

13) What is a protected field and when is it useful?

14) Which of the following can be instantiated?
    a. An Interface
    b. A Regular (concrete) Class
    c. A Base Class that extends from a Regular class

15) Are there situation(s) in which a class may not be instantiated? Explain.

16) What is meant by the following?
    a. Overloading
    b. Overriding
    c. Shadowing

17) What is the purpose of super(…) ??

18) What is the "substitutability" principle?