Revision Questions for Section D.1 - Objects as a programming concept

D.1.1 Nature of an object
_____Links: Object | Objects, Classes, Instances

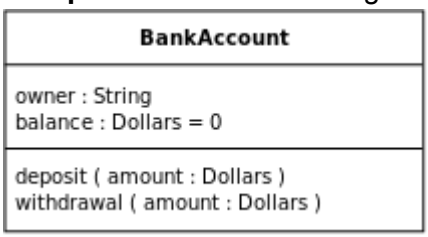
a. Outline the **definition** of an *object* in OOP in terms of its data and actions
b. List possible **data** and **actions** for: a person, car, fraction, date, music track, film.

D.1.2 Objects and Instantiation
_____Links: Instance | Objects, Classes, Instances

a. Distinguish between the *definition* of an object and the *instantiation* of an object.

D.1.3 UML Diagrams
D.1.4__Links: UML | Class Diagram

a. Describe the structure of a **UML** diagram as a means to represent an Object
b. **Construct** UML diagrams to represent the objects mentioned in D.1.1
c. **Interpret** the UML class diagram shown below:



D.1.5 Decomposition
_____Links: Decomposition | Algorithm Development

a. Describe the process of **decomposing** a problem into several related objects.
b. **Decompose** these examples into several related objects: employer, school, calculator, calendar, media collection.

D.1.6 Object Relationships
D.1.7__Links: Dependency | Aggregation | Inheritance | Inheritance and Class Hierarchy

a. Describe the **relationships** of *dependency*, *aggregation* and *inheritance* between objects.
b. **Link** these relationships to the operating terms: *'uses'*, '*has a*', and *'is a'*.
c. Outline the need to **reduce** dependencies between objects in OOP.

D.1.8 Construct Objects
_____Links: Object | Vehicle Example

a. Construct **three** related objects for: a school, a movie, a library.

D.1.9 Data Types
_____Links: Data Type | Integer | Real | String | Boolean | Types and Literals

a. Explain the need for different **types** of data to represent data items.
b. Give three **examples** of data that could be stored in these data types: *integer*, *real, string,*

Boolean

D.1.10 Parameters

Links: Parameter | Call by Value | Parameters

a. Define what is meant by a **parameter** in OOP.
b. Describe **how** data items can be passed to and from actions as parameters.
c. Explain what is meant by a *'pass by value'* parameter.