1. In object-oriented programming, what is meant by the following terms?

(a) Encapsulation [2]

Encapsulation is grouping data to one place, like one class

Wrapping attributes/variables/properties and methods/procedures/behaviours into a single entity (class)

Variables in the class are declared as private

(b) Data hiding [2]

Data hiding also groups data to one area, for example the same class, but prevents certain things in the class from accessing it

Hiding the details (attributes) of an object so that they cannot be changed by an outsider to reduce complexity like .sort() reduces complexity because you don’t need to know the code for the .sort() it just does it for you instead of having to manually write out the sort code yourself

(c) Instantiation [2]

Declaration of an instance

Creating a new object, given the class definition

(d) Inheritance [2]

The ability to access the parent class’s attributes and methods etc. For example an object of a subclass inherits it’s parent’s class’ attributes and methods and everything else the parent class has, meaning the child object can use it all too, except an inherited object you can also redefine certain instances, like if you want to redefine a method that was inherited

Ability of programming language of define subclasses in OOP which take on behaviours and attributes of a superclass (parent)

(e) Polymorphism [2]

Ability to have objects of different classes be treated in the same way

A programming languages ability to processs objects differently depending on their class

2. An object-oriented program is to be used to store and display details of members of a sports club.

A member class is defined which holds attributes surname, first name, annual membership fee, and methods to amend and display these details.

A junior member class inherits the attributes and methods of the member class but has an additional attribute date of birth.

An incomplete definition of the Member class is given below.

Complete the statements where indicated. [3]

class Member

private surname

private firstname

private annualFee

public procedure new(mySurname, myFirstName, myAnnualFee)

surname = mySurname

firstname = myFirstName

annualFee = myAnnualFee

endprocedure

public procedure amendDetails(mySurname, myfirstname, myAnnualFee)

*(leave this procedure incomplete)*

endprocedure

*(other procedures – do not complete)*

endclass

(a) Complete the definition of the JuniorMember class constructor. [2]

class JuniorMember inherits Member

private dateOfBirth

public procedure new(mySurname, myFirstname, myAnnualFee, myDateOfBirth)

super.new(mySurname, myFirstname, myAnnualFee)

dob= myDateOfBirth

endprocedure

*(other procedures – do not complete)*

endclass

(b) Write a statement to instantiate a junior member called Harry Mason, born 12/12/2004, annual fee £25.00. [1]

jmember1 = new JuniorMember(“Mason”, “Harry”, “25.00”, “12/12/2024”)

(c) Write a method which will amend the annual fee of a junior member. [2]

public procedure amendAnnualFee(newAnnualFee)

annualFee = newAnnualFee

endprocedure

(d) Write a method which will return the date of birth of a junior member. [2]

public procedure getDob()

return dob

endprocedure

[Total 20 marks]