9 **(a)** Describe what is meant by **an imperative (procedural)** programming language.

.....

.....

.....

..... [2]

(b) Describe what is meant by **a declarative** programming language.

.....

.....

.....

..... [2]

(c) Identify the programming paradigm for each of these program code examples.

Program code example	Programming paradigm

`male(john).`
`female(ethel).`
`parent(john, ethel).`	

`FOR Counter = 1 TO 20`
`  X = X * Counter`
`NEXT Counter`	

`Start: LDD Counter`
`  INC ACC`
`  STO Counter`	

`public class Vehicle`
`{`
`  private speed;`
`  public Vehicle()`
`  {`
`  speed = 0;`
`  }`
`}`	

[4]

2 Draw **one** line from each programming paradigm to its **most appropriate** description.

Programming paradigm	Description
Declarative	Programs using the instruction set of a processor
Imperative	Programs based on events such as user actions or sensor outputs
Low-level	Programs using the concepts of class, inheritance, encapsulation and polymorphism
Object-oriented	Programs with an explicit sequence of commands that update the program state, with or without procedure calls
	Programs that specify the desired result rather than how to get to it

[4]

2 A declarative language is used to represent the following facts about cats.

```
01 type(leopard, wild).
02 type(lion, wild).
03 type(cheetah, wild).
04 type(savannah, hybrid).
05 type(persian, domestic).
06
07 hair(leopard, medium).
08 hair(lion, short).
09 hair(cheetah, medium).
10 hair(savannah, medium).
11 hair(persian, long).
12
13 spots(leopard, yes).
14 spots(lion, no).
15 spots(cheetah, yes).
16 spots(savannah, yes).
17 spots(persian, no).
```

These clauses have the following meaning:

Clause	Meaning
01	A leopard is a type of wild cat.
08	A lion has short hair.
16	A savannah has spots.

(a) More facts are to be included. A **caracal** is a wild cat with short hair.

Write the additional clauses to record these facts.

18 .....

19 .....

[2]

(b) Using the variable `Cat`, the goal:

```
hair(Cat, medium)
```

returns

```
Cat = leopard, cheetah, savannah
```

Write the result returned by the goal:

```
hair(Cat, long)
```

`Cat` = ... [1]

(c) **(i)** Write the goal, using the variable `Pet`, to find all the domestic cats.

.....

..... [1]

(ii) Write the goal, using the variable `WildSpotty`, to find all the wild cats with spots.

.....

.....

.....

..... [2]

7 A program is to be written using Object-Oriented Programming (OOP) for a shop that sells knitting yarn. There are three types of yarn: acrylic, wool or mix.

The following data are stored for each type.

- Name
- Colour
- Batch code
- Weight
- Number of balls of yarn in stock (can be edited)
- Type of yarn

The following statements apply to yarn.

- Acrylic can be soft or not soft.
- Wool can be lamb, merino or alpaca.
- Mix contains a percentage of acrylic.

Each type of yarn has a method that will display all the information about the yarn.

(a) Complete this class inheritance diagram to show the **properties**, **methods** and **inheritance**.

Yarn
Name: STRING Colour: STRING BatchCode: STRING Weight: INTEGER NumberBalls: INTEGER Type: STRING
Constructor() EditNumberBalls() YarnInfo()

Acrylic
.....
Constructor() .....

Wool
.....
Constructor() .....

Mix
Percentage: INTEGER
Constructor() YarnInfo()

[5]

(b) Describe what is meant by the terms **properties**, **methods** and **inheritance**.

Properties .....

.....

.....

.....

Methods .....

.....

.....

.....

Inheritance .....

.....

.....

.....

[6]

11 **(a)** Define these Object-Oriented Programming (OOP) terms:

Instance .....

.....

Inheritance .....

.....

Polymorphism .....

.....

[3]

(b) In OOP, a class contains attributes and methods.

Complete the pseudocode for the class `Car` to enable objects to be created. The class needs to include:

- string attributes to store the make, model, body type and fuel type
- an integer attribute to store the number of cars of that type built.

The attributes must be available only through the methods of the class.

```
CLASS .....
    PRIVATE Make : STRING
    PRIVATE .....
    .....
    .....
    .....
    PUBLIC PROCEDURE New(CarMake : STRING, ..... ,
                        .....)
        Make ← .....
        Model ← .....
        BodyType ← CarBodyType
        Fuel ← ""
        NumberBuilt ← 0
    ENDPROCEDURE
    GetFuel()
    GetNumberBuilt()
.....
```

[5]