A A) • Fragment 1 - would most likely be the imperative paradigm. Features of this paradigm evident in the code fragment includes its declarative, uthlised the concept of forward chainning and is processor independent

• Fragment 2 — The paradigm of Functional would be the most likely paradigm. Features which are evident in code fragment too includes the use of variables, repitition (loop), assignment statements and mathematical concepts. This code fragment is similar to that of the language pascal which is apart of this paradigm.

B Reasons for the emergence of the object orientated paradigm (OOP) includes:

• Speed of code generation

• User Simplicity ie the ability of non computer programmer personnel having the ability to code simple projects.

• Allowing users the ability of modifing code modules / sub routines and behaviour of objects.

• Giving the programmer to focus & design the GUI in an

easy & precise way without the need to specify sizes (dimensions) colours and other attributes of screen elements in code

- Allows the re-use of existing modules / subroutines in an effective way and the ability of programmers to modify the behaviour of screen elements (objects) in an easy way in comparison to previous paradigms which required lengthly code to achieve the same effect resulting in an increase in productivity leading to an increase in code generation.

(c)(i) The logic error is that the while statement relies on the height entered during the program and the height has not been read yet by the computer.
This can be fixed by

- entering and reading the height before asking the WHILE statement.

OR

- Not including the while and adding an IF statement underneath height reading:

```
IF height <> 0
    THEN write('Enter width')
    . . .

ELSE stop program.
```

BOARD OF STUDIES
NEW SOUTH WALES

Question 24

Centre Number: ..... Student Number: .....

(00/11)

```
base:integer
function TTriangle.area:integer;
begin
    area:=(½*base*height);
end;
var
    InstTriangle:TTriangle;

begin
    write('Enter height');
    readln(InstTriangle.height);
    while InstTriangle.height <>0 do
    begin
        write('Enter width');
        readln(InstTriangle.width);
        writeln('The area is', InstTriangle.area);
    end;
    InstTriangle
end;
```

d) The logic paradigm would be best suited to this problem. The system described requires the software to make decisions based on the number of ~~passengers and~~ pieces of baggage required to go to any given sector. The logic paradigm allows the

Question 24 d) cont

computer to assess the load on particular chutes and decide how ~~chutes~~ chutes need to be available for each destination and class. This is because it is possible to input rules into the system such as, the number of bags a chute can handle and number of bags going to ~~a particular~~ the chutes. Using rules such as these, the system can determine which chutes will be ~~under~~ under heavy load and which ~~will~~ will be under a light ~~load~~ load. The system could then make a decision to allocate more or less chutes to different destinations and classes.