

User Defined Datatype



Papers Dock

COMPUTER SCIENCE 9618 PAPER 3

User Defined Datatype

Data types are like tools in a toolbox, each designed for a specific task in programming, dictating how data is stored and manipulated. Just as you wouldn't use a hammer to tighten a screw, each data type is suited to handle particular kinds of data (e.g., numbers, text) efficiently and appropriately.

What is meant by user defined datatype ?

- A data type constructed by a programmer and is not a primitive datatype (basic data types)
- A data type that references at least one other data type and is derived from one or more existing data type.
- The data types which are referenced can be primitive or user defined.
- To meet's programmer requirement

Explain why user defined datatype are necessary ?

- To create a new data type from existing data types.
- To allow data types not available in a programming language to be constructed that meets programmer's requirement

Datatypes

Non Composite	Composite
Single data type that does not refer to another data type	Data type that refers to other data types and are constructed from another datatype
Examples : Enumerated, Real, String, Char, Boolean, Integer	Examples : Record, List, Set, Array, Class, Queue, Linked List, Dictionary

Composite data types are like toolkits, combining multiple tools (e.g., lists, arrays, structs) to hold and manage collections of related data, whereas **non-composite (primitive) data types** are like single tools (e.g., int, float, char) designed to handle one specific type of data.

Composite types bundle these individual tools together to solve more complex tasks, while non-composite types focus on simpler, specific tasks.

Explain what is meant by Composite Data Type ?

- A user defined data type that is a collection of data that can consist of multiple elements of different or the same data types.
- grouped under a single identifier

Explain what is meant by Non Composite Data Type ?

- It can be defined without referencing another data type.
- It can be primitive type available in a programming language, or a user-defined datatype.

Enumerated Datatypes

A user defined non composite data type with a list of all possible values that is ordered

Pseudocode To Declare Enumerated Datatype

TYPE name of Datatype = (___, ___, ___, ___)

Declare an enumerated data type with the identifier Months which can have all 12 months.

TYPE Months = (January , February , March, April,)

Declare a variable currentmonth of Datatype Months

DECLARE currentmonth : Months

Assign August in currentmonth variable

currentmonth <--- August

Declare a variable previousmonth of Datatype Month

DECLARE previousmonth : Months

Print the previous month

previousmonth <--- currentmonth - 1

OUTPUT previousmonth

Exam Style Question

- (b) A user-defined data type, timeOfDay, is declared using the following pseudocode.

```
TYPE timeOfDay = (morning, afternoon, evening, night)
```

- (i) Identify the type of user-defined data type declared and state its classification.

Type **Enumerated**

Classification **Non - Composite**

[2]

- (ii) Write pseudocode to declare the variable session of type timeOfDay.
Assign the value afternoon to the variable session.

DECLARE session : timeOfDay

session <--- afternoon

[2]

- (a) Write pseudocode to create an enumerated type called Parts to include these parts sold in a computer shop:

Monitor, CPU, SSD, HDD, LaserPrinter, Keyboard, Mouse

TYPE Parts = (Monitor, CPU, SSD, HDD, LaserPrinter, Keyboard, Mouse)

[2]

3(a)

One mark for each marking point (Max 2)

2

- **TYPE Parts =**
- **(Monitor, CPU, SSD, HDD, LaserPrinter, Keyboard, Mouse)**

Complete answer

TYPE Parts = (Monitor, CPU, SSD, HDD, LaserPrinter, Keyboard, Mouse)

- 4 Two descriptions of user-defined data types are given.

Give appropriate type declaration statements for each, including appropriate names.

- (a) A data type to hold a set of prime numbers below 20. These prime numbers are:

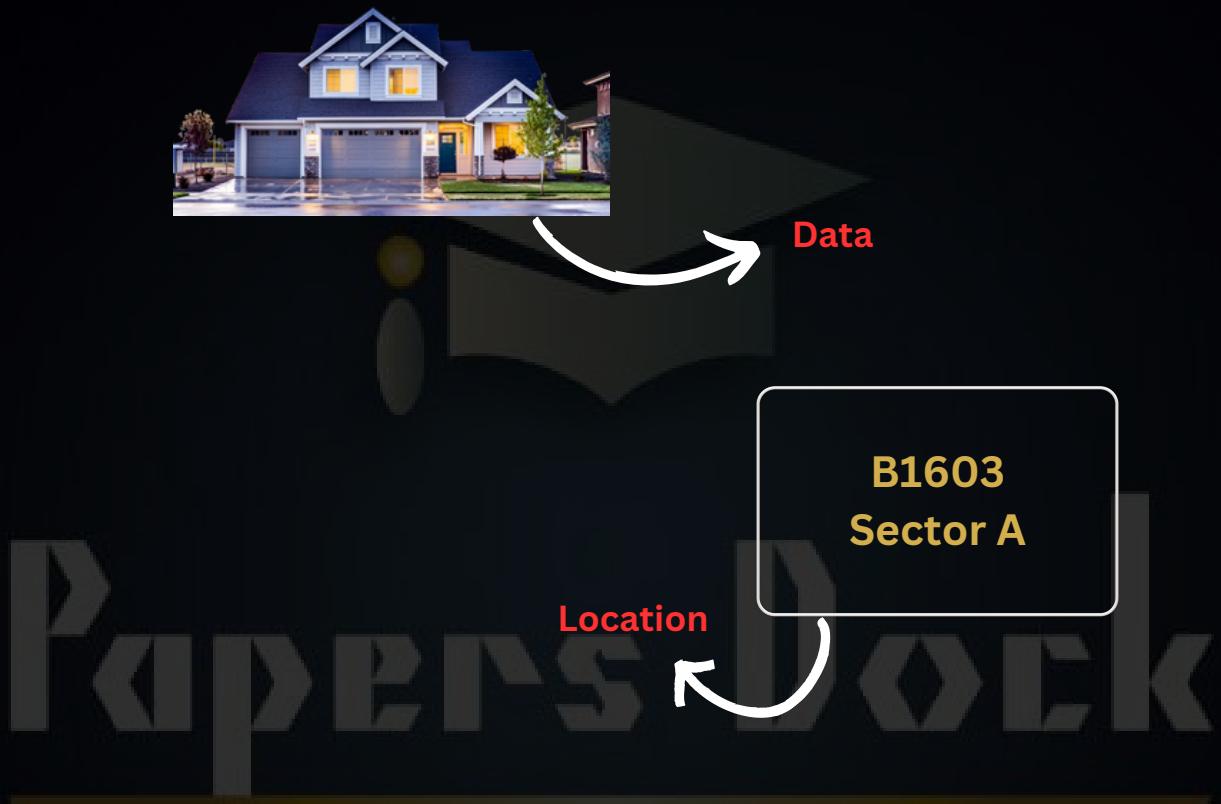
2, 3, 5, 7, 11, 13, 17, 19

TYPE Prime = (2, 3, 5, 7, 11, 13, 17)

[2]

Pointer Datatypes

A user defined non composite datatype that stores memory locations only and indicates the type of data stored in the memory location.



A pointer is like the address of a house. Just as the address tells you where the house is located, a pointer tells you where a variable's value is stored in memory. Instead of holding the house (data) itself, the pointer holds the address (memory location) where you can find the house (data).

we are going to make a variable to store the location 8217. So that variable would need a specific datatype known as pointer.

Memory Location	Content
8216	
8217	“Ahmed”
8218	
8219	

Base String

Pseudocode To Declare Pointer Datatype

^ : This symbol represents pointer

@ : This symbol represents the address is required not the data

TYPE name of Datatype = ^Base Datatype

Depends on the data stored so it could be String, Integer, Real, Boolean, Char

Question : We have Integer variables so create a data type with the name IntegerPointer

TYPE IntegerPointer = ^INTEGER

Declare a variable in which you store the address of integer values with the name myintegerpointer

DECLARE myintegerpointer : IntegerPointer

Question : You have a variable Number with value 10 in it stored on 5216 location. Store the address in myintegerpointer variable

myintegerpointer <---- @Number

(address of number not the value)

DEREFENCING

You have the address and you want the value on that address

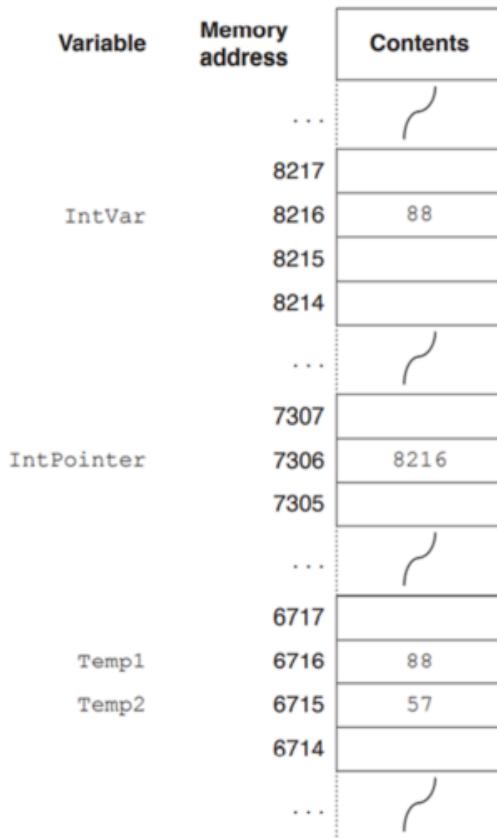
Change the data which is currently pointed by myintegerpointer to 100

myintegerpointer ^ <--- 100

- (c) A pointer is a variable that stores the address of a variable of a particular type.

Consider the code on page 3, which uses the following identifiers:

Identifier	Data type	Description
IntPointer	$^{\wedge}$ INTEGER	pointer to an integer
IntVar	INTEGER	an integer variable
Temp1	INTEGER	an integer variable
Temp2	INTEGER	an integer variable



Use the diagram to state the current values of the following expressions:

(i) $@Temp2$ [1]

(ii) IntPointer [1]

(iii) $IntPointer^{\wedge}$ [1]

(iv) $IntPointer^{\wedge} = Temp2 + 6$ [1]

- (d) Write pseudocode statements that will achieve the following:

(i) Assign the value 22 to the variable Temp2.

..... [1]

(ii) Place the address of Temp1 in IntPointer.

..... [1]

(iii) Copy the value in Temp2 into the memory location currently pointed at by IntPointer.

..... [1]

```

Sum ← 91           // assigns the value 91 to the integer variable Sum
IPointer ← @Sum    // assigns to IPointer the address of the
                    // integer variable Sum
MyInt1 ← IPointer^ // assigns to variable MyInt1 the value at an
                    // address pointed at by IPointer
IPointer^ ← MyInt2 // assigns the value in the variable MyInt2 to
                    // the memory location pointed at by IPointer

```

The four assignment statements are executed. The diagram shows the memory contents after execution.

Variable	Memory Address	Contents
IPointer	5848	
	5847	
	5846	4402
	5845	
	... 4403	
Sum	4402	33
	4401	
	... 3428	
	3427	91
	3426	33
MyInt1	3425	
	... 3428	
	3427	91
MyInt2	3426	33
	3425	
	... 5848	

Use the diagram to state the current values of the following expressions:

- (i) IPointer [1]
- (ii) IPointer^ [1]
- (iii) @MyInt1 [1]
- (iv) IPointer^ = MyInt2 [1]

(d) Write pseudocode statements that will achieve the following:

- (i) Place the address of MyInt2 in IPointer.

..... [1]

- (ii) Assign the value 33 to the variable MyInt1.

..... [1]

- (iii) Copy the value in MyInt2 into the memory location currently pointed at by IPointer.

..... [1]

Record Datatypes

It's a user defined datatype composite data type and a group of multiple data types

Emp No	Name	Age	Department	Salary
001	Alex S	26	Store	5000
002	Golith K	32	Marketing	5600

If you want to store a complete record in a single variable name we use the concept of record datatype

Pseudocode To Declare Record Datatype

```
TYPE name of Datatype  
DECLARE Value1 : Datatype  
DECLARE Value2 : Datatype  
ENDTYPE
```

Question : We need to store information of a book under a single identifier. Create a record datatype with the name Book. The book should hold info about ISBN number (Integer), Title (String), Genre (String).

TYPE Book

DECLARE ISBN : INTEGER

DECLARE Title : STRING

DECLARE Genre : STRING

ENDTYPE

Question : Declare a new record variable named MyBook

DECLARE MyBook : Book

Question : The value of ISBN is 124657, Title is “Papersdock”, Genre is “Fiction”

MyBook.ISBN <--- 124657

MyBook.Title <--- “Papersdock”

MyBook.Genre <--- “Fiction”

Question : Why Pointer is Non - Composite

Because the value which is stored in a variable as location is not an integer or string or any other data type.

- 1 Consider the following user-defined data type.

```
TYPE Book
    DECLARE ISBN      : INTEGER
    DECLARE Author   : STRING
    DECLARE Title    : STRING
    DECLARE Supplier : (Amazone, Stones, Smiths, Blackwalls, Greens,
                        Coals, Boarders)
ENDTYPE
```

- (a) Name the data type of Book.

..... [1]

- (b) Name the non-composite data type used in the Supplier declaration.

..... [1]

- (c) (i) Write a pseudocode statement to declare a variable, BestSeller, of type Book.

..... [1]

- (ii) Write a pseudocode statement to assign "John Williams" to the author of BestSeller.

..... [1]

- 1 (a) Consider the following user-defined data type:

```
TYPE LibraryBookRecord
    DECLARE ISBN      : INTEGER
    DECLARE Title    : STRING
ENDTYPE
```

- (i) Write a pseudocode statement to declare a variable, Book, of type LibraryBookRecord.

..... [1]

- (ii) Write a pseudocode statement that assigns 'Dune' to the Title of Book.

..... [1]

Declaring A Range

Question : Declare a variable named Number which contains 0 till 99 numbers

DECLARE Number : 0 .. 9

Specific Datatype In Array

If Array is declared with the data type String that means any string value can be stored in the array but if you want to make it specific values only then you are suppose to specify the values that could be assign to elements.

Question : Declare an Array that can only have “Taha” “Bano” “Pappan”

DECLARE Names : Array [1:3] OF (“Taha”, “Bano”, “ Pappan”)

Note : If there are more than one thing to store than declare array and use Specific Datatype (with Quotation Marks) Average always means real values. If you have single value then you would need Enumerated (without Quotation Marks). Enumerated types are ideal for situations where you have a limited set of related values.

(b) The user-defined data type `LibraryBookRecord` needs to be modified by adding the following fields:

- a field called `Genre` which can take two values, fiction or non-fiction
- a field called `NumberOfLoans` which can be an integer value in the range 1 to 99

Write the updated version of `LibraryBookRecord`.

.....
.....
.....
.....
.....
.....
.....

[3]

1 Data types can be defined using pseudocode.

The data type, `LibraryRecord`, is defined in pseudocode as:

```
TYPE LibraryRecord
    DECLARE Title : STRING
    DECLARE Fiction : BOOLEAN
    DECLARE Author : STRING
    DECLARE NumberOfCopies : INTEGER
ENDTYPE
```

A variable, `LibraryBook`, is declared in pseudocode as:

```
DECLARE LibraryBook : LibraryRecord
```

(a) Write **pseudocode** statements to assign:

- A Level Computer Science to `Title` of `LibraryBook`
- FALSE to `Fiction` of `LibraryBook`.

.....
.....
.....
.....

[2]

(b) The type definition for `LibraryRecord` is changed.

(i) The value for `NumberOfCopies` must be between 1 and 10 inclusive.

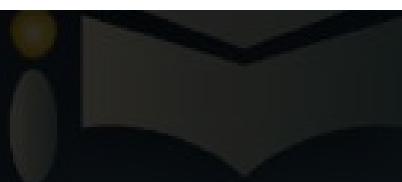
Write the updated line of **pseudocode** from the type definition of `LibraryRecord` to implement the change.

..... [1]

(ii) Every copy of every book is now uniquely identified by an accession number, `AccessionNumber`, as it is added to the library. Each library record will include one or more accession numbers. Each accession number is an integer.

Write the extra line of **pseudocode** needed in the type definition of `LibraryRecord`.

.....
.....
.....
..... [2]



Papers Dock

Set Datatypes

The SET is a user-defined composite data type that allows you to define a collection of elements of a specific data type, where each element is unique.

TYPE Name Of Set = SET OF data type

DEFINE Name of datatype (val1, val2 , val3) : Name Of Set

TYPE LetterSet = SET OF CHAR

DEFINE Vowels ('A','E','I','O','U'): LetterSet

Exam Style Question

- (b) Write **pseudocode** statements to declare the set data type EvenNumbers to hold this set of even numbers between 2 and 12:

2, 4, 6, 8, 10, 12

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

[4]

3(b)

One mark per mark point (**Max 4**)

MP1 Type statement fully correct

MP2 DEFINE EvenNumbers

MP3 Correct list of values in brackets

MP4 : <set identifier> from Type statement used

Example answer

TYPE Numbers = SET OF INTEGER

DEFINE EvenNumbers (2, 4, 6, 8, 10, 12): Numbers

Write pseudocode statements to declare a set data type HexDigits that holds the set of valid hexadecimal digit characters: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

```
TYPE HexSet = SET OF CHAR
DEFINE Hexadecimal ('0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A',
'B', 'C', 'D', 'E', 'F') : HexSet
```



User Defined Data types

Question 1

- 2 Data types can be classified as composite or non-composite.

A record is declared of type `box` using the following pseudocode.

```
TYPE size = (small, medium, large)

TYPE box

    DECLARE volume : size

    DECLARE price : REAL

    DECLARE colour : STRING

ENDTYPE

DECLARE myBox : ARRAY [1:6] OF box
```

- (a) (i) Identify **one** composite and **three** non-composite data types used in the pseudocode.

Composite data type

Non-composite data type 1

Non-composite data type 2

Non-composite data type 3

[4]

- (b) A box is red, with medium volume and a price of \$10.99.

Write **pseudocode** to store the details of this box in the first element of the array.

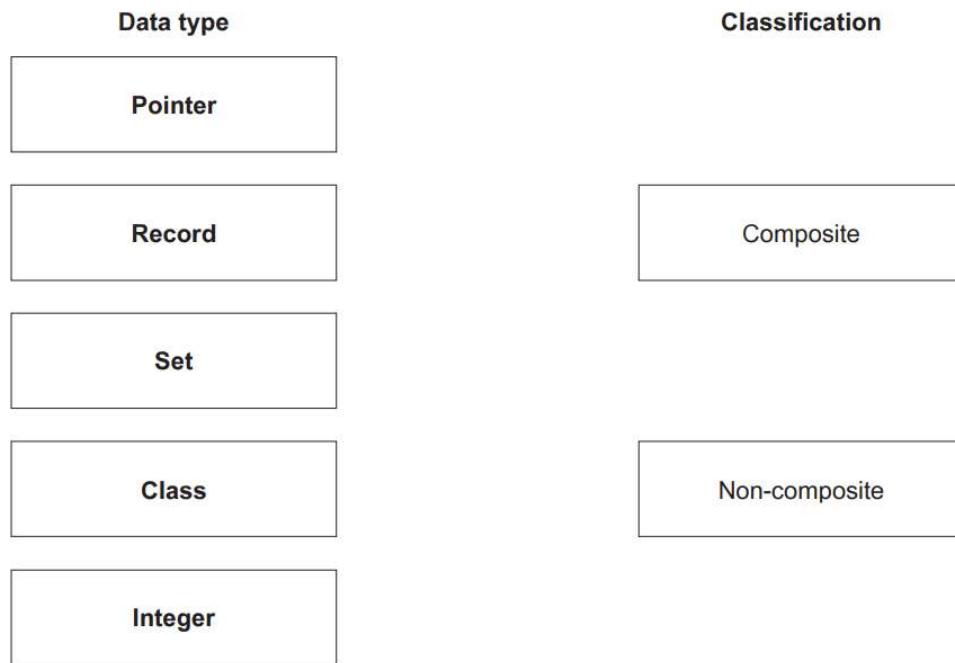
```
.....  
.....  
.....  
.....  
.....  
.....
```

[3]

Question 2

2 Data types can be classified as composite or non-composite.

(a) Draw **one** line from each data type to its correct classification.



[2]

(b) A user-defined data type, timeOfDay, is declared using the following pseudocode.

```
TYPE timeOfDay = (morning, afternoon, evening, night)
```

(i) Identify the type of user-defined data type declared **and** state its classification.

Type

Classification

[2]

(ii) Write pseudocode to declare the variable session of type timeOfDay.
Assign the value afternoon to the variable session.

.....

.....

.....

.....

[2]

Question 3

- 2 Data types can be classified as composite or non-composite.

A record is declared of type `box` using the following pseudocode.

```
TYPE size = (small, medium, large)

TYPE box

    DECLARE volume : size
    DECLARE price : REAL
    DECLARE colour : STRING

ENDTYPE

DECLARE myBox : ARRAY [1:6] OF box
```

- (a) (i) Identify **one** composite and **three** non-composite data types used in the pseudocode.

Composite data type

Non-composite data type 1

Non-composite data type 2

Non-composite data type 3

[4]

- (ii) Identify the data type in the pseudocode that is enumerated.

..... [1]

- (b) A box is red, with medium volume and a price of \$10.99.

Write **pseudocode** to store the details of this box in the first element of the array.

.....
.....
.....
.....
..... [3]

Question 4

- 5 (a) Explain why user-defined data types are necessary.

.....
.....
.....
.....

[2]

- (b) An organisation stores data about its employees.

- Employee ID is a five-digit number, for example, 01234.
- Employee name is a string, for example, 'Kiri Moana'.
- Department is one of three values: Sales, Technical, Customer services.
- Salary is an integer value in the range 25 000 to 150 000.

- (i) Complete the following **pseudocode** definition of a user-defined data type to store the employee data.

```
TYPE Employee  
  
DECLARE EmployeeID : .....  
  
DECLARE EmployeeName : STRING  
  
DECLARE Department : ( .....  
.....)  
  
DECLARE Salary : 25000..150000
```

[4]

- (ii) Write a **pseudocode** statement to declare a variable, NewEmployee of data type Employee.

.....
.....

[1]

- (iii) Write a **pseudocode** statement that assigns 02244 to the EmployeeID of NewEmployee.

.....
.....

[1]

- (iv) Employee is an example of a record that is a composite data type.

State **two** other composite data types.

1

2

[2]

Question 5

i

- 6 (a) State what is meant by a **user-defined data type**.

.....
..... [2]

- (b) A pseudocode declaration for a user-defined data type for the months of the year is as follows:

```
TYPE
    DECLARE Months: (January, February, March, April, May, June, July,
                      August, September, October, November, December)
ENDTYPE
```

- (i) Identify this type of user-defined data type.

.....
..... [1]

- (ii) Write a **pseudocode** statement to declare a variable CurrentMonth of data type Months.

.....
..... [1]

- (iii) Write a **pseudocode** statement to assign the value August to the variable CurrentMonth.

.....
..... [1]

Question 6

2 A programmer uses non-composite and composite data types to create a program.

(a) Define the term **non-composite data type**.

.....
.....

[1]

(b) Describe **two** different non-composite data types.

Data type 1

Description

.....
.....

Data type 2

Description

.....
.....

[4]

(c) Define the term **composite data type**.

.....
.....

[1]

(d) Describe **two** different composite data types.

Data type 1

Description

.....
.....

Data type 2

Description

.....
.....

[4]

Question 7

- 1 Data types can be defined in a programming language.

The data type, StudentRecord, is defined by the code:

```
TYPE StudentRecord
    DECLARE StudentID      : INTEGER
    DECLARE StudentFirstName : STRING
    DECLARE StudentSurname   : STRING
    DECLARE StudentDOB       : DATE
    DECLARE StudentCourse    : ARRAY[1:10] OF STRING
ENDTYPE
```

A variable, CollegeStudent, is declared with the code:

```
DECLARE CollegeStudent : StudentRecord
```

- (a) Write a pseudocode statement to assign 6539 to the StudentID of CollegeStudent.

..... [1]

- (b) The type definition for StudentRecord is changed.

- (i) Students can take six courses from: Computer Science, Engineering, Science, Maths, Physics, Chemistry, Music, Drama and English Language.

Rewrite one line from the type definition of StudentRecord to implement the change.

```
DECLARE .....
```

.....
.....
.....
.....

[2]

- (ii) The values for the field StudentID must be between 1 and 8000 inclusive.

Rewrite one line from the type definition of StudentRecord to implement the change.

```
DECLARE .....
```

[1]

- (c) A programmer is asked to write a program to process the assessment data for each student. Students sit one exam in every course they take.

A composite data type, `StudentAssessment`, needs to be defined with the following three fields.

- a student assessment code (a unique code of three letters and two digits)
- the marks for the six exams
- the average mark of the six exams

- (i) Write **pseudocode** to define the data type `StudentAssessment`.

.....
.....
.....
.....
.....
.....
.....
.....

[4]

- (ii) Data about all students and their assessments are stored in a file that uses random organisation. The `StudentID` is used as the key field.

The program allows a user to enter data for a new student.

Explain how the program adds the new data to the file.

.....
.....
.....
.....
.....
.....
.....
.....

[3]

Question 8

- 1 Consider the following user-defined data type.

```
TYPE Book
    DECLARE ISBN      : INTEGER
    DECLARE Author   : STRING
    DECLARE Title    : STRING
    DECLARE Supplier : (Amazone, Stones, Smiths, Blackwalls, Greens,
                        Coals, Boarders)
ENDTYPE
```

- (a) Name the data type of Book.

.....[1]

- (b) Name the non-composite data type used in the Supplier declaration.

.....[1]

- (c) (i) Write a pseudocode statement to declare a variable, BestSeller, of type Book.

.....[1]

- (ii) Write a pseudocode statement to assign "John Williams" to the author of BestSeller.

.....[1]

Question 9

- 1 (a) Consider the following user-defined data type:

```
TYPE LibraryBookRecord
    DECLARE ISBN      : INTEGER
    DECLARE Title    : STRING
ENDTYPE
```

- (i) Write a pseudocode statement to declare a variable, Book, of type LibraryBookRecord.

.....[1]

- (ii) Write a pseudocode statement that assigns 'Dune' to the Title of Book.

.....[1]

(b) The user-defined data type `LibraryBookRecord` needs to be modified by adding the following fields:

- a field called `Genre` which can take two values, fiction or non-fiction
- a field called `NumberOfLoans` which can be an integer value in the range 1 to 99

Write the updated version of `LibraryBookRecord`.

.....
.....
.....
.....
.....
.....
.....
.....

[3]

(c) A pointer is a variable that stores the address of a variable of a particular type.

Consider the code on page 3, which uses the following identifiers:

Identifier	Data type	Description
<code>IntPointer</code>	<code>^INTEGER</code>	pointer to an integer
<code>IntVar</code>	<code>INTEGER</code>	an integer variable
<code>Temp1</code>	<code>INTEGER</code>	an integer variable
<code>Temp2</code>	<code>INTEGER</code>	an integer variable

```
IntVar ← 57           // assigns the value 57 to the integer
                     // variable IntVar
IntPointer ← @IntVar // assigns to IntPointer the address of the
                     // integer variable IntVar
Temp2 ← IntPointer^ // assigns to variable Temp2 the value at an
                     // address pointed at by IntPointer
IntPointer^ ← Temp1 // assigns the value in the variable Temp1 to
                     // the memory location pointed at by IntPointer
```

The four assignment statements are executed. The diagram shows the memory contents after execution.

Variable	Memory address	Contents
IntVar	8217	
	8216	88
	8215	
	8214	
	...	
IntPtr	7307	
	7306	8216
	7305	
	...	
	6717	
Temp1	6716	88
Temp2	6715	57
	6714	
	...	

Use the diagram to state the current values of the following expressions:

- (i) @Temp2 [1]
- (ii) IntPtr [1]
- (iii) IntPtr^ [1]
- (iv) IntPtr^ = Temp2 + 6 [1]

(d) Write pseudocode statements that will achieve the following:

- (i) Assign the value 22 to the variable Temp2.
..... [1]
- (ii) Place the address of Temp1 in IntPtr.
..... [1]
- (iii) Copy the value in Temp2 into the memory location currently pointed at by IntPtr.
..... [1]

Question 10

- 1 (a) Consider the following pseudocode user-defined data type:

```
TYPE MyContactDetail  
    DECLARE Name : STRING  
    DECLARE HouseNumber : INTEGER  
ENDTYPE
```

- (i) Write a pseudocode statement to declare a variable, NewFriend, of type MyContactDetail.

..... [1]

- (ii) Write a pseudocode statement that assigns 129 to the HouseNumber of NewFriend.

..... [1]

- (b) The user-defined data type MyContactDetail needs to be modified by:

- adding a field called Area which can take three values, uptown, downtown or midtown
- amending the field HouseNumber so that house numbers can only be in the range 1 to 499.

Write the updated version of MyContactDetail.

.....
.....
.....
.....
.....
.....
.....
.....

[3]

- (c) A pointer is a variable that stores the address of a variable of a particular type.

Consider the pseudocode on page 3, which uses the following identifiers:

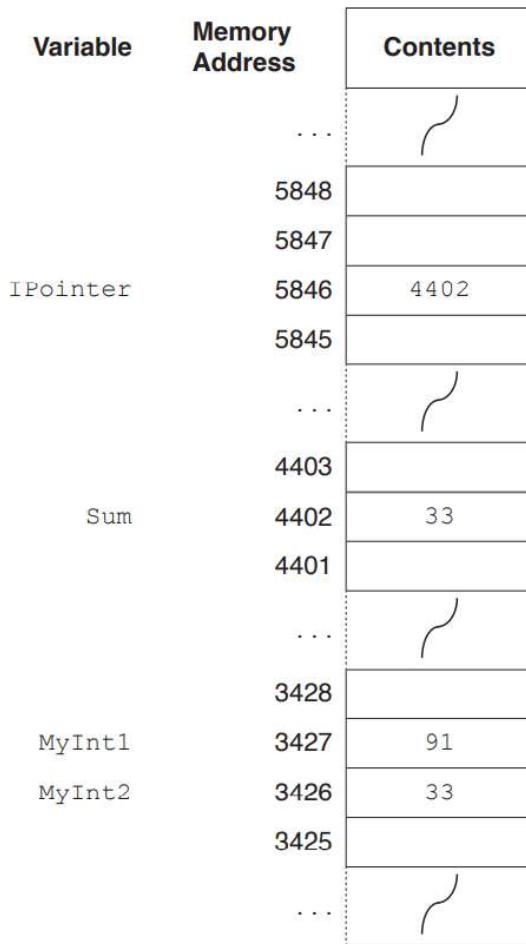
Identifier	Data type	Description
IPointer	^INTEGER	pointer to an integer
Sum	INTEGER	an integer variable
MyInt1	INTEGER	an integer variable
MyInt2	INTEGER	an integer variable

```

Sum ← 91           // assigns the value 91 to the integer variable Sum
IPointer ← @Sum    // assigns to IPointer the address of the
                    // integer variable Sum
MyInt1 ← IPointer^ // assigns to variable MyInt1 the value at an
                    // address pointed at by IPointer
IPointer^ ← MyInt2 // assigns the value in the variable MyInt2 to
                    // the memory location pointed at by IPointer

```

The four assignment statements are executed. The diagram shows the memory contents after execution.



Use the diagram to state the current values of the following expressions:

- (i) IPointer [1]
- (ii) IPointer^ [1]
- (iii) @MyInt1 [1]
- (iv) IPointer^ = MyInt2 [1]

(d) Write pseudocode statements that will achieve the following:

- (i) Place the address of `MyInt2` in `IPointer`.

.....[1]

- (ii) Assign the value 33 to the variable `MyInt1`.

.....[1]

- (iii) Copy the value in `MyInt2` into the memory location currently pointed at by `IPointer`.

.....[1]

Question 11

3 (a) A particular programming language allows the programmer to define their own data types.

`ThisDate` is an example of a user-defined structured data type.

```
TYPE ThisDate
    DECLARE ThisDay      : (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
                           13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
                           24, 25, 26, 27, 28, 29, 30, 31)
    DECLARE ThisMonth   : (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug,
                           Sep, Oct, Nov, Dec)
    DECLARE ThisYear    : INTEGER
ENDTYPE
```

A variable of this new type is declared as follows:

```
DECLARE DateOfBirth : ThisDate
```

- (i) Name the non-composite data type used in the `ThisDay` and `ThisMonth` declarations.

.....[1]

- (ii) Name the data type of `ThisDate`.

.....[1]

- (iii) The month value of `DateOfBirth` needs to be assigned to the variable `MyMonthOfBirth`.

Write the required statement.

.....[1]

- (b)** Annual rainfall data from a number of locations are to be processed in a program.

The following data are to be stored:

- location name
- height above sea level (to the nearest metre)
- total rainfall for each month of the year (centimetres to 1 decimal place)

A user-defined, composite data type is needed. The programmer chooses `LocationRainfall` as the name of this data type.

A variable of this type can be used to store all the data for one particular location.

- (i)** Write the definition for the data type `LocationRainfall`.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

[5]

Question 12

- 4 (a)** A particular programming language allows the programmer to define their own data types.

An example of a user-defined data type for an address is:

```
TYPE ThisAddress
    DECLARE ThisHouseNo : INTEGER
    DECLARE ThisStreet  : STRING
    DECLARE ThisTown   : STRING
ENDTYPE
```

A variable of this new type is declared as follows:

```
DECLARE HomeAddress : ThisAddress
```

- (i)** Write the statement that assigns the house number 34 to `HomeAddress`.

.....[1]

(ii) The type definition for `ThisAddress` is to be changed.

Rewrite one line from the definition for each of the following changes.

House numbers are in the range from 1 to 10.

DECLARE

The possible towns are limited to: Brightown, Arunde and Shoram.

DECLARE [2]

(b) Temperature data from a number of weather stations are to be processed by a program.

The following data are to be stored:

- weather station ID (a unique four-letter code)
- latitude (to 2 decimal places)
- average temperature (to the nearest whole number) for each year from 2001 to 2015 inclusive

A programmer designs a composite data type `WeatherStation`. A variable of this type can be used to store all the data for one particular station.

(i) Write the definition for the user-defined data type `WeatherStation`.

.....
.....
.....
.....
.....
.....
.....

[5]

Question 13

- 2 (a) Describe the purpose of a user-defined data type.

.....
.....
.....
.....
..... [2]

- (b) Define, using pseudocode, the following enumerated data types:

- (i) SchoolDay to hold data about the days students are usually in school.

.....
..... [1]

- (ii) WeekEnd to hold data about the days that are not school days.

.....
..... [1]

- (c) Define, using pseudocode, the composite data type ClubMeet. This will hold data about club members that includes:

- first name and last name
- the two days they attend:
 - one on a school day
 - one not on a school day.

Use the enumerated types you created in part (b).

.....
.....
.....
.....
.....
.....
.....
.....
.....
..... [4]

Question 14

3 Enumerated and pointer are two non-composite data types.

- (a) Write **pseudocode** to create an enumerated type called `Parts` to include these parts sold in a computer shop:

Monitor, CPU, SSD, HDD, LaserPrinter, Keyboard, Mouse

.....
.....
.....

[2]

- (b) Write **pseudocode** to create a pointer type called `SelectParts` that will reference the memory location in which the current part name is stored.

.....
.....
.....

[2]

Question 15

1 Data types can be defined using pseudocode.

The data type, `LibraryRecord`, is defined in pseudocode as:

```
TYPE LibraryRecord
    DECLARE Title : STRING
    DECLARE Fiction : BOOLEAN
    DECLARE Author : STRING
    DECLARE NumberOfCopies : INTEGER
ENDTYPE
```

A variable, `LibraryBook`, is declared in pseudocode as:

```
DECLARE LibraryBook : LibraryRecord
```

- (a) Write **pseudocode** statements to assign:

- A Level Computer Science to Title of `LibraryBook`
- FALSE to Fiction of `LibraryBook`.

.....
.....
.....
.....

[2]

(b) The type definition for LibraryRecord is changed.

(i) The value for NumberOfCopies must be between 1 and 10 inclusive.

Write the updated line of **pseudocode** from the type definition of LibraryRecord to implement the change.

.....
.....

[1]

(ii) Every copy of every book is now uniquely identified by an accession number, AccessionNumber, as it is added to the library. Each library record will include one or more accession numbers. Each accession number is an integer.

Write the extra line of **pseudocode** needed in the type definition of LibraryRecord.

.....
.....
.....
.....

[2]

(c) A record is a user-defined composite data type.

Explain what is meant by a **user-defined composite data type**.

Include an example of **another** user-defined composite data type in your answer.

.....
.....
.....
.....
.....

[3]

Question 16

- 1 Data types can be defined using pseudocode.

The data type, BuildingRecord, is defined in pseudocode as:

```
TYPE BuildingRecord
    DECLARE BuildingID : INTEGER
    DECLARE BuildingGroup : STRING
    DECLARE OwnerName : STRING
    DECLARE BuildingAddress : STRING
    DECLARE DateLastSold : DATE
    DECLARE PriceLastSold : REAL
ENDTYPE
```

A variable, BuildingRegister, is declared in pseudocode as:

```
DECLARE BuildingRegister : BuildingRecord
```

- (a) Write **pseudocode** statements to assign:

- 1067 to the BuildingID of BuildingRegister
- house to the BuildingGroup of BuildingRegister

.....
.....
.....
.....

[2]

- (b) The type definition for BuildingRecord is changed. The data type for BuildingGroup is changed to an enumerated type, BuildingType, with values of house, bungalow, apartment and farm.

- (i) Write the type declaration for BuildingType in **pseudocode**.

.....
.....
.....
.....

[2]

- (ii) Write the new declaration for BuildingGroup in **pseudocode**.

.....
.....

[1]

- (iii) Write the new **pseudocode** statement to assign house to BuildingGroup of BuildingRegister.

.....
.....

[1]

- (c) The program is to be rewritten using Object-Oriented Programming (OOP). The data type BuildingRecord is to be changed to a class, BuildingClass.

The properties for BuildingClass are BuildingID, BuildingGroup, OwnerName, BuildingAddress, DateLastSold and PriceLastSold.

All the properties are set to PRIVATE, for example:

```
PRIVATE PriceLastSold : REAL
```

- (i) Write the declaration in **pseudocode** for OwnerName as PRIVATE.

..... [1]

- (ii) Explain why the properties have been set to PRIVATE.

.....
.....
.....
..... [2]

Question 17

- 3 (a) State what is meant by the term **enumerated data type**.

..... [1]

- (b) State what is meant by the term **pointer data type**.

..... [1]

- (c) The months of the year are: January, February, March, April, May, June, July, August, September, October, November and December.

Write the **pseudocode** statement to define the data type Quarter1, to hold the names of the first three months of a year.

.....
.....
.....
..... [2]

- (d) The composite data type `Pet` is used to store data about the various pets of a group of students. It uses these fields:

Field name	Data type
PetName	String
AnimalType	String
PetAge	Integer
PetGender	Char
OwnerName	String

- (i) Write the **pseudocode** statement to set up a variable for one record of the composite data type `Pet`.

.....
..... [1]

- (ii) Write **pseudocode** to store the details of the following pet, in the variable you set up in part (d)(i).

PetName	AnimalType	PetAge	PetGender	OwnerName
Tibbles	Cat	8	M	Jasmine Smith

.....
.....
.....
.....
.....
..... [3]

Question 18

- 4 A program to manage regular flight details at an airport requires some user-defined data types.
- (a) Write **pseudocode** statements to declare the enumerated data type Aircraft to hold data about the types of aircraft used for a flight.

These types of aircraft are: C300, C350, D242, E757, X380.

.....
.....
.....
..... [2]

- (b) Write **pseudocode** statements to declare the composite data type Flight to hold data about flights to a specific destination. These include:
- flight number, which could be any combination of letters and numbers
 - destination
 - date of departure
 - type of aircraft used.

Use the enumerated data type you created in part (a).

.....
.....
.....
.....
.....
.....
.....
.....
..... [4]

- (c) (i) Write the **pseudocode** statement to set up a variable for one record of the composite data type Flight.

..... [1]

- (ii) Write **pseudocode** to store the details of the following flight in the variable you set up in part (c)(i).

Field	Data
flight number	XA782
destination	Cambridge
date of departure	12/12/2022
type of aircraft used	C350

Use the field names you created in part (b).

.....
.....
.....
.....
.....
..... [3]

Question 19

- 4 Two descriptions of user-defined data types are given.

Give appropriate type declaration statements for each, including appropriate names.

- (a) A data type to hold a set of prime numbers below 20. These prime numbers are:

2, 3, 5, 7, 11, 13, 17, 19

.....
.....
.....
..... [2]

- (b) A data type to point to a day in the week, for example Monday.

.....
.....
.....
..... [2]

Question 20

6 (a) Write **pseudocode** statements to declare the composite data type, `TAppointments`, to hold data about patients for a dental clinic. It will include for each patient:

- name (first name and last name)
- date of birth
- telephone number
- date of last appointment
- date of next appointment
- all treatments are complete (yes or no).

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

[4]

Question 21

3 Describe what is meant by **enumerated** and **pointer** data types.

Enumerated

.....
.....
.....
.....

Pointer

.....
.....
.....
.....

[4]

Question 22

- 2 Describe what is meant by **composite** and **non-composite** data types.

Composite

.....

.....

Non-composite

.....

.....

.....

[4]

Answer

Answer 1

2(a)(i)	Composite box Non-composite size / enumerated REAL STRING	4
2(a)(ii)	size	1
2(b)	myBox[1].volume ← medium myBox[1].price ← 10.99 myBox[1].colour ← "red"	3

Answer 2

2(a)	2 marks for all 5 single lines correct 1 mark for 4 lines correct otherwise zero		2
	Data type	Classification	
	Pointer		
	Record	Composite	2
	Set		
	Class	Non-composite	
	Integer		
2(b)(i)	Type Classification	Enumerated Non-composite	2
2(b)(ii)	DECLARE session : timeOfDay session ← afternoon		2

Answer 3

2(a)(i)	Composite box Non-composite size / enumerated REAL STRING	4
2(a)(ii)	size	1
2(b)	myBox[1].volume ← medium myBox[1].price ← 10.99 myBox[1].colour ← "red"	3

Answer 4

5(a)	<p>1 mark per bullet point to max 2</p> <ul style="list-style-type: none"> • No suitable data type is provided by the language used • The programmer needs specify a new data type • ... that meets the requirements of the application / program 	2
5(b)(i)	<p>1 mark per bullet point</p> <ul style="list-style-type: none"> • EmployeeID declared as STRING • Sales, Technical and CustomerServices ... • ... with commas in-between • ENDTYPE <pre>TYPE Employee DECLARE EmployeeID : STRING DECLARE EmployeeName : STRING DECLARE Department : (Sales, Technical, CustomerServices) DECLARE Salary : 25000..150000 ENDTYPE</pre>	4
5(b)(ii)	DECLARE NewEmployee : Employee	1
5(b)(iii)	NewEmployee.EmployeeID ← "02244"	1
5(b)(iv)	<p>1 mark per bullet point to max 2</p> <ul style="list-style-type: none"> • Array • List • Set • Collection • Class • Stack • Queue • Linked list • Dictionary 	2

Answer 5

6(a)	<p>1 mark per bullet point to max 2</p> <ul style="list-style-type: none"> • Derived from one or more existing data types • Used to extend the built-in data types • Creates data-types specific to applications // programmer's requirements 	2
6(b)(i)	Enumerated (data type)	1
6(b)(ii)	DECLARE CurrentMonth : Months	1
6(b)(iii)	CurrentMonth ← August	1

Answer 6

2(a)	<u>single data type</u> that does not involve a reference to another type/usually built in to a programming language	1
2(b)	1 mark for data type, 1 for definition, max 4 , 2 data types <ul style="list-style-type: none"> oo Integer oo Stores a whole number oo Boolean oo Stores true or false/1 or 0/on or off oo Real/Single/Double/Float/Decimal oo Stores decimal numbers oo String oo Stores zero or more characters oo Char oo Stores a single character oo Pointer oo Whole number used to reference a memory location 	4
2(c)	data type constructed from other data types	1
2(d)	1 mark for naming, 1 for description, max 4 , 2 data types <ul style="list-style-type: none"> oo Record oo collection of related items which may have different data types oo Array oo (Indexed) collection of items with the same data type oo List oo (Indexed) collection of items that can have different data types oo Set oo stores a finite number of different values that have no order // supports mathematical operations oo Class/Structure oo Gives the properties and methods for an object 	4

Answer 7

1(a)	CollegeStudent.StudentID ← 6539	1
1(b)(i)	1 mark per bullet <ul style="list-style-type: none"> • StudentCourse: ARRAY[1:6] OF • All valid string options , for example: DECLARE StudentCourse: ARRAY[1:6] OF ("Computer Science", "Engineering", "Science", "Maths", "Physics", "Chemistry", "Music", "Drama", "English Language") 	2
1(b)(ii)	DECLARE StudentID: 1 .. 8000	1
1(c)(i)	1 mark per bullet <ul style="list-style-type: none"> • Type declaration TYPE and ENDTYPE • Declaring Code as STRING • Declaring Mark as ARRAY [1:6] OF INTEGER • AverageMark as REAL <p>For example:</p> <pre>TYPE StudentAssessment DECLARE Code : STRING DECLARE Mark : ARRAY[1:6] OF INTEGER DECLARE AverageMark : REAL ENDTYPE</pre>	4

1(c)(ii)	Any 3 from, 1 mark per bullet <ul style="list-style-type: none"> • StudentID/key field is hashed to produce home location • If home location is free, insert record/data • Else use overflow method to find free location to store record / data • If no free location available then file is full and record/data cannot be stored 	3
----------	---	----------

Answer 8

1	(a) Record	1
	(b) Enumerated	1
	(c) DECLARE BestSeller : Book	1
	(d) BestSeller.Author ← "John Williams"	1

Answer 9

1(a)(i)	DECLARE Book : LibraryBookRecord	1
1(a)(ii)	Book.Title ← "Dune"	1
1(b)	<pre>TYPE LibraryBookRecord DECLARE ISBN : INTEGER DECLARE Title : STRING DECLARE Genre : (Fiction, Non-Fiction) DECLARE NumberOfLoans : 1 .. 99 ENDTYPE</pre> <p>mark for correct declaration and first two fields (note: only if attempt at modification)</p>	3
1(c)(i)	6715	1
1(c)(ii)	8216	1
1(c)(iii)	88	1
1(c)(iv)	FALSE	1
1(d)(i)	Temp2 ← 22	1
1(d)(ii)	IntPtr ← @Temp1	1
1(d)(iii)	IntPtr^ ← Temp2	1

Answer 10

1(a)(i)	DECLARE NewFriend : MyContactDetail	1	
1(a)(ii)	NewFriend.HouseNumber ← 129	1	
1(b)	Declaration of Name, Area, HouseNumber Inclusion of three correct values for Area Inclusion of correct range for HouseNumber	1 1 1 For example: TYPE MyContactDetail DECLARE Name : STRING DECLARE Area : (uptown, downtown, midtown) DECLARE HouseNumber : 1..499 ENDTYPE	3
1(c)(i)	4402	1	
1(c)(ii)	33	1	
1(c)(iii)	3427	1	
1(c)(iv)	TRUE	1	
1(d)(i)	IPointer ← @MyInt2	1	
1(d)(ii)	MyInt1 ← 33	1	
1(d)(iii)	IPointer^ ← MyInt2	1	

Answer 11

3 (a) (i)	enumerated	1
(ii)	record	1
(iii)	MyMonthOfBirth ← DateOfBirth.ThisMonth	1
(b) (i)	TYPE LocationRainfall DECLARE LocationName : STRING DECLARE LocationHeight : INTEGER DECLARE TotalMonthlyRainfall : ARRAY[1..12] OF REAL ENDTYPE	1 1 1 1 1+1

Answer 12

4 (a) (i)	HomeAddress.ThisHouseNo ← 34	1
(ii)	DECLARE ThisHouseNo: 1..10 DECLARE ThisTown: [Brightown, Arunde, Shoram]	1 1
(b) (i)	TYPE WeatherStation DECLARE StationID : STRING DECLARE Latitude : REAL DECLARE Temperature : ARRAY[1..15] OF INTEGER ENDTYPE	1 1 1 1+1 1

Answer 13

Question	Answer	Marks
2(a)	<p>One mark for each correct marking point (Max 2)</p> <ul style="list-style-type: none"> • To create a new data type (from existing data types) • To allow data types not available in a programming language to be constructed // To extend the flexibility of the programming language 	2
2(b)(i)	TYPE SchoolDay = (Monday, Tuesday, Wednesday, Thursday, Friday)	1
2(b)(ii)	TYPE WeekEnd = (Saturday, Sunday)	1
2(c)	<p>One mark for each marking point (Max 4)</p> <ul style="list-style-type: none"> • TYPE ClubMeet and ENDTYPE correct • DECLARE FirstName and DECLARE LastName included with correct data types • DECLARE Schoolday included with correct data types from part 2(b)(i) • DECLARE Weekend included with correct data types from part 2(b)(ii) <p>Example answer</p> <pre>TYPE ClubMeet DECLARE FirstName : STRING DECLARE LastName : STRING DECLARE Schoolday : SchoolDay DECLARE Weekend : WeekEnd ENDTYPE</pre>	4

Answer 14

3(a)	<p>One mark for each marking point (Max 2)</p> <ul style="list-style-type: none"> • TYPE Parts = • (Monitor, CPU, SSD, HDD, LaserPrinter, Keyboard, Mouse) <p>Complete answer</p> <pre>TYPE Parts = (Monitor, CPU, SSD, HDD, LaserPrinter, Keyboard, Mouse)</pre>	2
3(b)	<p>One mark for each marking point (Max 2)</p> <ul style="list-style-type: none"> • TYPE SelectParts = ^ • correct data type chosen Parts <p>Complete answer</p> <pre>TYPE SelectParts = ^Parts</pre>	2

Answer 15

1(a)	LibraryBook.Title ← "A Level Computer Science" LibraryBook.Fiction ← FALSE
1(b)(i)	DECLARE NumberOfCopies : 1 .. 10
1(b)(ii)	DECLARE AccessionNumber : ARRAY[1:NumberOfCopies] OF INTEGER
1(c)	Any two from <ul style="list-style-type: none">• A data type constructed by a programmer // not a primitive data type• A data type that references at least one other data type...• ... the data types can be primitive, or user defined One mark for an example <ul style="list-style-type: none">• Class / object / set

Answer 16

1(a)	BuildingRegister.BuildingID ← 1067 BuildingRegister.BuildingGroup ← "house"
1(b)(i)	One mark: TYPE BuildingType = One mark: (house, bungalow, apartment, farm) TYPE BuildingType = (house, bungalow, apartment, farm)
1(b)(ii)	DECLARE BuildingGroup : BuildingType
1(b)(iii)	BuildingRegister.BuildingGroup ← house
1(c)(i)	PRIVATE OwnerName : STRING
1(c)(ii)	To ensure that attributes can only be accessed by the class's own methods To enforce encapsulation // ensure they are hidden

Answer 17

3(a)	A (user-defined non-composite) data type with an ordered list of possible values.
3(b)	A user-defined non-composite data type used to reference a memory location .
3(c)	Marks as shown in the square brackets: TYPE Quarter1 = (January, February, March) TYPE Quarter1 = [1] (January, February, March) [1]
3(d)(i)	DECLARE Pet1 : Pet

Answer 18

4(a)	Marks as shown in the square brackets: TYPE Aircraft = (C300, C350, D242, E757, X380) TYPE Aircraft = [1] (C300, C350, D242, E757, X380) [1]
4(b)	One mark for each point (Max 4) <ul style="list-style-type: none">• TYPE Flight and ENDTYPE correct• DECLARE FlightNumber and DECLARE Destination as STRING• DECLARE DepartureDate as DATE• DECLARE AircraftType as Aircraft (correct data type from part 4(a)) <p>Example answer:</p> <pre>TYPE Flight DECLARE FlightNumber : STRING DECLARE Destination : STRING DECLARE DepartureDate : DATE DECLARE AircraftType : Aircraft ENDTYPE</pre>
4(c)(i)	Example answer: DECLARE Flight1 : Flight
4(c)(ii)	One mark for each point (Max 3) <ul style="list-style-type: none">• Correct assignments of both string data values• Correct assignments of date data value• Correct assignments of enumerated data value <p>Example answer:</p> <pre>Flight1.FlightNumber ← "XA782" Flight1.Destination ← "Cambridge" Flight1.DepartureDate ← 12/12/2022 Flight1.AircraftType ← C350</pre>

Answer 19

4(a)	<p>One mark per mark point (Max 2)</p> <ul style="list-style-type: none"> • TYPE Prime • = (2, 3, 5, 7, 11, 13, 17, 19) <p>Example answer</p> <pre>TYPE Prime = (2, 3, 5, 7, 11, 13, 17, 19)</pre>
4(b)	<p>One mark per mark point (Max 2)</p> <ul style="list-style-type: none"> • TYPE TDayPointer • = ^STRING // ^DayOfWeek <p>Example answer</p> <pre>TYPE TDayPointer = ^STRING // ^DayOfWeek</pre>

Answer 20

6(a)	<p>One mark for TYPE TAppointments and ENDTYPE correct One mark for every two correct declarations (Max 3)</p> <p>Example answer</p> <pre>TYPE TAppointments DECLARE Name : STRING DECLARE DateOfBirth : DATE DECLARE Telephone : STRING DECLARE LastAppointment : DATE DECLARE NextAppointment : DATE DECLARE TreatmentsComplete : BOOLEAN ENDTYPE</pre>	4
------	---	---

Answer 21

3	<p>One mark per mark point – enumerated type (Max 2)</p> <p>MP1 A user-defined non-composite (data type) (only award once) MP2 ...with a list of all possible values MP3 ...that is ordered.</p> <p>One mark per mark point – pointer type (Max 2)</p> <p>MP4 A user-defined non-composite (data type) (only award once) MP5 ...that stores addresses/memory locations only MP6 ...and indicates the type of data stored in the memory location.</p>	4
---	--	---

Answer 22

2	<p>One mark per mark point – composite (Max 2)</p> <p>MP1 A (user defined) data type that is a collection of data that can consist of multiple elements MP2 ...of different or the same data types MP3 ...grouped under a single identifier.</p> <p>One mark per mark point – non-composite (Max 2)</p> <p>MP4 It can be defined without referencing another data type. MP5 It can be a primitive type available in a programming language, or a user-defined type.</p>	4
---	--	---