








3.1.1 User-defined data types(Pastpapers 2015 – 2019)

-  show understanding of why user-defined types are necessary
-  define and use non-composite types: enumerated, pointer
-  define and use composite data types: set, record and class/object
-  choose and design an appropriate user-defined data type for a given problem

3.1.2 File organisation and access

-  show understanding of methods of file organisation: serial, sequential (using a key field) and random (using a record key)
-  show understanding of methods of file access: – sequential access for serial and sequential files – direct access for sequential and random files
-  select an appropriate method of file organisation and file access for a given problem

9608/32/M/J/15

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]

(ii) The programmer decides to store all the data in a file. Initially, data from 27 locations will be stored. More rainfall locations will be added over time and will never exceed 100.

The programmer has to choose between two types of file organisation. The two types are serial and sequential.

Give two reasons for choosing **serial** file organisation.

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

Answers
9608/32/M/J/15

3	(a) (i)	enumerated	1
	(ii)	record	1
	(iii)	MyMonthOfBirth ← DateOfBirth.ThisMonth	1
(b)	(i)	<pre> TYPE LocationRainfall DECLARE LocationName : STRING DECLARE LocationHeight : INTEGER DECLARE TotalMonthlyRainfall : <u>ARRAY[1..12]</u> OF REAL ENDTYPE </pre>	1 1 1 1+1
	(ii)	<ul style="list-style-type: none"> no need to re-sort data every time new data is added only a small file so searching will require little processing new records can easily be appended 	1 1 1 [max 2]
			Total: 10



9608/33/M/J/15

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]



(ii) The programmer decides to store all the data in a file. The number of weather stations could grow to reach 20000, but not all stations will be present at first.

The programmer decides on random organisation for the file.

Describe three steps which show how a new weather station record is added to the file.

- 1
- 2
- 3[3]

Answers
9608/33/M/J/15

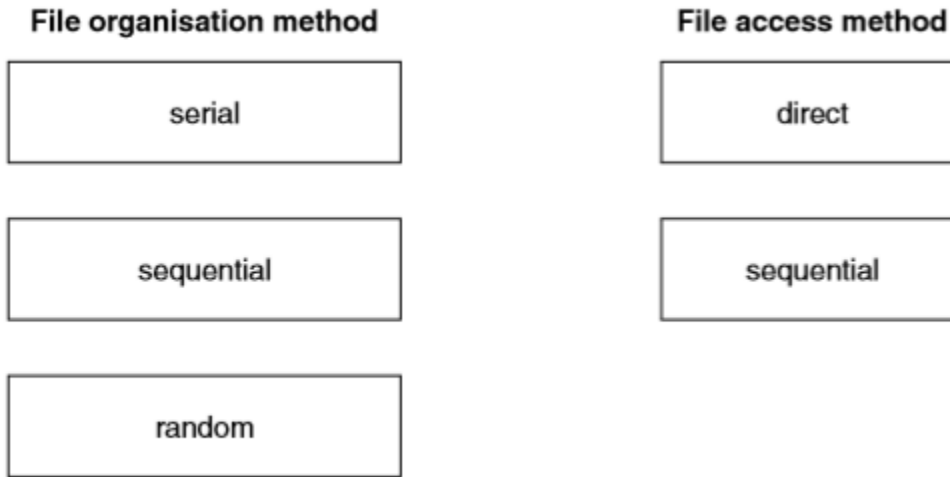
4 (a)	(i) HomeAddress.ThisHouseNo ← 34	1
	(ii) DECLARE ThisHouseNo: 1..10 DECLARE ThisTown: [Brightown, Arunde, Shoram]	1 1
(b)	(i) TYPE WeatherStation DECLARE StationID : STRING DECLARE Latitude : REAL DECLARE Temperature : <u>ARRAY[1..15]</u> OF INTEGER ENDTYPE	1 1 1 + 1 1
	(ii) StationID is hashed to produce home location If home location is free insert record Else use overflow method to find free location	1 1 1
		Total: 11



9608/31/M/J/16




4 (a) Three file organisation methods and two file access methods are shown below.

Draw lines to link each file organisation method to its appropriate file access method or methods






[4]

(b) A bank has a very large number of customers. The bank stores data for each customer. This includes:

-  unique customer number
-  personal data (name, address, telephone number)
-  transactions

The bank computer system makes use of three files:

-  A – a file that stores customer personal data. This file is used at the end of each month for the production of the monthly statement
-  B – a file that stores encrypted personal identification numbers (PINs) for customer bank cards. This file is accessed when the customer attempts to withdraw cash at a cash machine (ATM).
-  C – a file that stores all customer transaction records for the current month. Every time the customer makes a transaction, a new record is created.

For each of the files A, B and C, state an appropriate method of organisation. Justify your choice.

(i) File A organisation

Justification

.....

.....

.....[3]

(ii) File B organisation
Justification
.....
.....[3]

(iii) File C organisation
Justification
.....
.....[3]

Answers
9608/31/M/J/16

4 (a)	<table border="0"> <tr> <td style="text-align: center;">File organisation method</td> <td></td> <td style="text-align: center;">File access method</td> </tr> <tr> <td style="text-align: center;">serial</td> <td style="text-align: center;">↗</td> <td style="text-align: center;">direct</td> </tr> <tr> <td style="text-align: center;">sequential</td> <td style="text-align: center;">↘</td> <td style="text-align: center;">sequential</td> </tr> <tr> <td style="text-align: center;">random</td> <td style="text-align: center;">↘</td> <td style="text-align: center;">sequential</td> </tr> </table>	File organisation method		File access method	serial	↗	direct	sequential	↘	sequential	random	↘	sequential	1 2 1
File organisation method		File access method												
serial	↗	direct												
sequential	↘	sequential												
random	↘	sequential												
(b) (i)	<p>Sequential As all customers get statement ... // high hit rate Suitable for batch processing of the records // the records will be processed one after the other File organised using customer's unique ID (as primary key field) // Serial As all customers get statement ... // high hit rate Suitable for batch processing of the records // the records will be processed one after the other Order not important</p>	1 1 1 1 1 1 1 1 Max 3												
(ii)	<p>Random Real-time transaction processing Requires fastest access to data No need to search through records</p>	1 1 1 1 Max 3												



9608/31/M/J/17

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
IntPointer	^INTEGER	pointer to an integer
IntVar	INTEGER	an integer variable
Temp1	INTEGER	an integer variable
Temp2	INTEGER	an integer variable

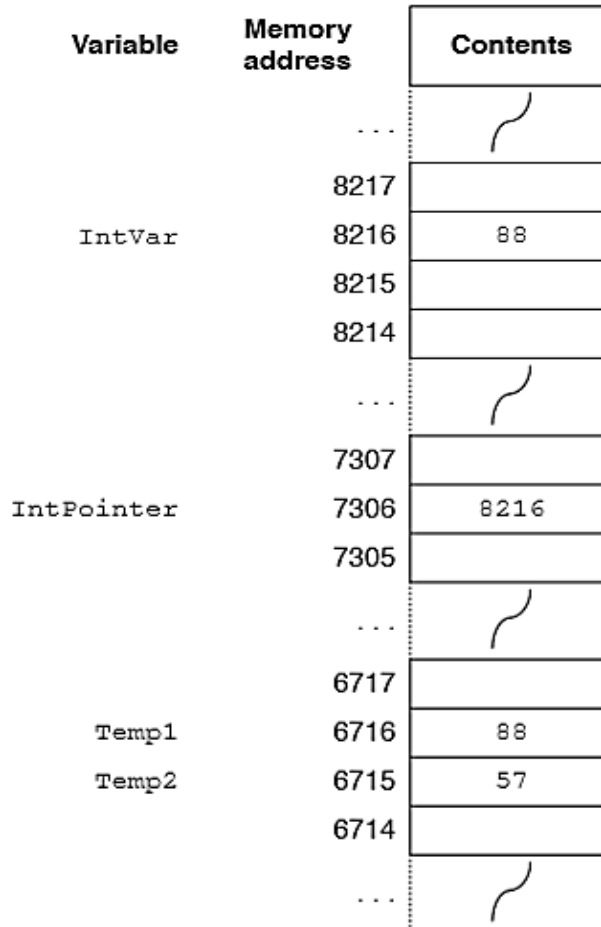


```

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

```

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) @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 IntPointer.

.....[1]

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

.....[1]

9608/31/M/J/17

Question	Answer	Marks
1(a)(i)	DECLARE Book : LibraryBookRecord	1
1(a)(ii)	Book.Title ← "Dune"	1
1(b)	TYPE LibraryBookRecord DECLARE ISBN : INTEGER DECLARE Title : STRING DECLARE Genre : (Fiction, Non-Fiction) DECLARE NumberOfLoans : 1 .. 99 ENDTYPE mark for correct declaration and first two fields (note : only if attempt at modification)	3 1 1 1
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)	IntPointer ← @Temp1	1
1(d)(iii)	IntPointer^ ← Temp2	1

9608/32/M/J/17

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**.

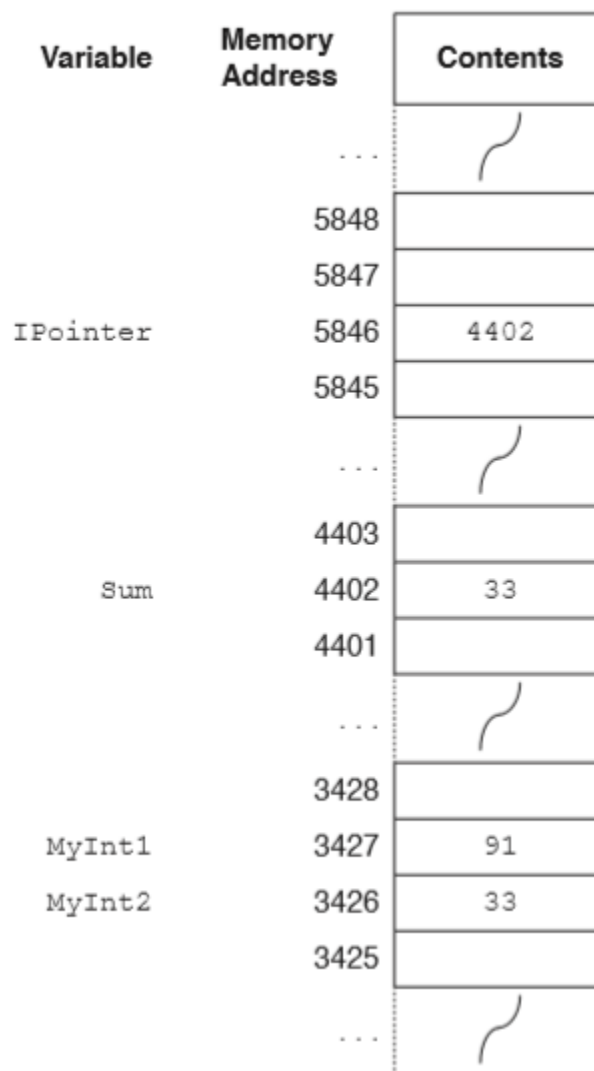
.....
.....
.....
.....
.....
.....
.....[3]

(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
<code>IPointer</code>	<code>^INTEGER</code>	pointer to an integer
<code>Sum</code>	<code>INTEGER</code>	an integer variable
<code>MyInt1</code>	<code>INTEGER</code>	an integer variable
<code>MyInt2</code>	<code>INTEGER</code>	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]

Answers
9608/32/M/J/17

Question	Answer	Marks
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 For example: TYPE MyContactDetail DECLARE Name : STRING DECLARE Area : (uptown, downtown, midtown) DECLARE HouseNumber : 1..499 ENDTYPE	1 1 1 1 1 & 1
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



9608/31/M/J/18

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]



Answers

9608/31/M/J/18

Question	Answer	Marks
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 <input type="checkbox"/> Integer <input type="checkbox"/> Stores a whole number <input type="checkbox"/> Boolean <input type="checkbox"/> Stores true or false/1 or 0/on or off <input type="checkbox"/> Real/Single/Double/Float/Decimal <input type="checkbox"/> Stores decimal numbers <input type="checkbox"/> String <input type="checkbox"/> Stores zero or more characters <input type="checkbox"/> Char <input type="checkbox"/> Stores a single character <input type="checkbox"/> Pointer <input type="checkbox"/> Whole number used to reference a memory location	4
2(c)	data type constructed from other data types	1

Question	Answer	Marks
2(d)	1 mark for naming, 1 for description, max 4 , 2 data types <input type="checkbox"/> Record <input type="checkbox"/> collection of related items which may have different data types <input type="checkbox"/> Array <input type="checkbox"/> (Indexed) collection of items with the same data type <input type="checkbox"/> List <input type="checkbox"/> (Indexed) collection of items that can have different data types <input type="checkbox"/> Set <input type="checkbox"/> stores a finite number of different values that have no order // supports mathematical operations <input type="checkbox"/> Class/Structure <input type="checkbox"/> Gives the properties and methods for an object	4

9608/32/M/J/18

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]

**Answers
9608/32/M/J/18**



1(a)	<code>CollegeStudent.StudentID ← 6539</code>	1
1(b)(i)	1 mark per bullet <ul style="list-style-type: none">• <code>StudentCourse: ARRAY[1:6] OF</code>• All valid string options , for example: <code>DECLARE StudentCourse: ARRAY[1:6] OF ("Computer Science", "Engineering", "Science", "Maths", "Physics", "Chemistry", "Music", "Drama", "English Language")</code>	2
1(b)(ii)	<code>DECLARE StudentID: 1 .. 8000</code>	1
1(c)(i)	1 mark per bullet <ul style="list-style-type: none">• Type declaration <code>TYPE</code> and <code>ENDTYPE</code>• Declaring <code>Code</code> as <code>STRING</code>• Declaring <code>Mark</code> as <code>ARRAY [1:6] OF INTEGER</code>• <code>AverageMark</code> as <code>REAL</code> For example: <code>TYPE StudentAssessment</code> <code> DECLARE Code : STRING</code> <code> DECLARE Mark : ARRAY[1:6] OF INTEGER</code> <code> DECLARE AverageMark : REAL</code> <code>ENDTYPE</code>	4
1(c)(ii)	Any 3 from, 1 mark per bullet <ul style="list-style-type: none">• <code>StudentID/key</code> 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