Past-paper walk-through

Program Development Life cycle ▪ 12.1

eXercise

Questions in this set

- 9618/22 N25 Q5 ▪ 5 marks
- 9618/21 N24 Q5 ▪ 10 marks
- 9618/23 N24 Q5 ▪ 7 marks

Reading the mark scheme

- **B1** a mark that stands on its own – the point is either made or not.
- **M1** a *method* mark: the right approach, even if the number is wrong.
- **A1** an *answer* mark, and it usually needs its M mark first.
- **C1** a working mark on the way to the answer.
- // separates answers the examiner accepts equally.
- **ecf** = error carried forward: a later part is marked on your own earlier answer.
- **owtte** = "or words to that effect" – the wording need not match.

Question 5

9618/22 N25 ▪ Q5 ▪ 5 marks

A program is developed to satisfy a specific customer requirement.
The project follows a program development life cycle model. This model divides the development process into several different stages.
(a) The table lists some of the development activities.
Complete the table by writing the name of the life cycle stage for each activity:

Activity	**Name of life cycle stage**
a structure chart is produced	
a program is modified to allow it to run on new hardware	
the programmer identifies the customer's requirements	
an Integrated Development Environment (IDE) provides context-sensitive help such as 'auto-complete'	

Question 5

[4]

Question 5

Activity	Name of life cycle stage
a structure chart is produced	
a program is modified to allow it to run on new hardware	
the programmer identifies the customer's requirements	
an Integrated Development Environment (IDE) provides context-sensitive help such as 'auto-complete'	

[4]

Question 5

9618/22 N25 ▪ Q5 ▪ 5 marks

A program is developed to satisfy a specific customer requirement.

The project follows a program development life cycle model. This model divides the development process into several different stages.

(b) Alpha and beta testing has been completed. The final testing stage is carried out by the customer.

Identify this final testing stage. **[1]**

Solution 5

(a) Mark scheme

- Activity
- Name of life cycle stage a structure chart is produced
- Design a program is modified to allow it to run on new hardware
- Maintenance the programmer identifies the customer's requirements
- Analysis an Integrated Development Environment (IDE) provides context-sensitive help such as 'auto- complete'
- Coding
- One mark per row

Solution 5

(b)

Mark scheme

- Acceptance testing
- 1
- October/November
- 2025

Question 5

9618/21 N24 ▪ Q5 ▪ 10 marks

A software developer follows a program development life cycle. The life cycle divides the development process into various stages.

(a) The following table lists some development activities.

Complete the table by writing the name of the life cycle stage for each activity.

Activity	**Name of life cycle stage**
The walkthrough method is used.	
An algorithm is implemented in a programming language.	
The client is interviewed about problems with the current system.	
The program is modified to run on new hardware.	
Records and file structures are defined.	

Question 5

[5]

Question 5

Activity	Name of life cycle stage
The walkthrough method is used.	
An algorithm is implemented in a programming language.	
The client is interviewed about problems with the current system.	
The program is modified to run on new hardware.	
Records and file structures are defined.	

[5]

Question 5

Type of test data	Test data value	Expected result
Normal	30	TRUE

Question 5

9618/21 N24 ▪ Q5 ▪ 10 marks

A software developer follows a program development life cycle. The life cycle divides the development process into various stages.

(b)(i) The function will:

- take an integer value as a parameter
- return `TRUE` if the value is within the range 24 to 37, inclusive
- otherwise return `FALSE`.

Complete the table to define a test plan to thoroughly test the operation of the function.

Question 5

Type of test data	Test data value	Expected result
Normal	30	`TRUE`

Question 5

[4]

(b)(ii) The function is to be tested on its own. When it is shown to work correctly the function will be combined with other modules and testing will continue.

Identify the type of testing that this represents. **[1]**

Solution 5

(a) Worked steps

Match each activity to the stage where it actually happens, not to the stage it sounds like.

Activity	Stage
The walkthrough method is used	Testing
An algorithm is implemented in a programming language	Coding
The client is asked what the system must do	Analysis
A structure chart is produced	Design
The finished system is handed over and monitored	Maintenance

Solution 5

The two that catch people out: a walkthrough is a **testing** technique even though no code runs, and a structure chart belongs to **design**, before any coding. **Mark scheme**

- Activity
- Name of life cycle stage
- The walkthrough method is used.
- Testing
- An algorithm is implemented in a programming language.
- Coding
- The client is interviewed about problems with the current system.
- Analysis
- The program is modified to run on new hardware.
- Maintenance
- Records and file structures are defined.

Solution 5

- Design
- One mark per row

Solution 5

(b)(i) Worked steps

Test data is chosen to probe a **boundary**, not to confirm what already works. One row per type:

Type	Value	Expected result
Abnormal	12	FALSE
Boundary	23	TRUE
Boundary	22	FALSE
Normal	30	TRUE

Solution 5

- **Boundary** data comes in pairs straddling the limit — 22 and 23 — because that is where a > written for a >= shows up.
- **Abnormal** data is a value the function should reject; **normal** is an ordinary accepted value.
- Every row needs its expected result stated, and the result must follow the specification rather than the code.

Solution 5

Mark scheme

- One mark per row
- Type of test data
- Test data value
- Expected result
- Abnormal
- 12 (< 23)
- FALSE
- Abnormal / Boundary / Extreme
- 23
- FALSE
- Boundary / Extreme
- 24

Solution 5

- TRUE
- Boundary
- 25
- TRUE
- Boundary
- 36
- TRUE
- Boundary /Extreme
- 37
- TRUE
- Abnormal / Boundary / Extreme
- 38
- FALSE
- Abnormal

Solution 5

- 99 (> 38)
- FALSE
- Max 4 marks

Solution 5

(b)(ii)

Mark scheme

- Integration (testing)

Question 5

9618/23 N24 ▪ Q5 ▪ 7 marks

5 A software developer follows a program development life cycle. The life cycle divides the development process into various stages.

(a) The following table lists some development activities.

Complete the table by writing the name of the life cycle stage for each activity.

Activity	Name of life cycle stage
A compiler is used.	
A program that has been released for general use is modified.	
The dry run method is used.	
The program structure is specified.	

[4]

(b) A software developer has written modules `Test A()` and `Test B()`. These have been

Question 5

written but contain errors. These modules are called from several places in the main program and testing of the main program (integration testing) has to stop.

Identify a method that can be used to continue testing the main program **before** the errors in these modules have been corrected **and** describe how this would work.

Question 5

Question 5

[3]

Solution 5

(a) Mark scheme

- Activity
- Name of life
- cycle stage
- A compiler is be used.
- Coding
- A program that has been released for general use
- is modified
- Maintenance

Solution 5

- The dry run method is used.
- Testing
- The program structure is specified.
- Design
- One mark per row
- 4

Solution 5

(b) Worked steps

One mark for the **method** and two for the **description**, so name it and then say how it works.

- The method is **stub testing**.
- The modules `Test_A()` and `Test_B()` are **replaced by dummy modules** — stubs ...
- ... which **return a known result**, or display an output statement when called, so the caller can be tested and its behaviour checked.

Solution 5

The reason it exists is worth stating: a module can then be tested **before the modules it calls have been written**, which is what makes top-down development possible at all. Without stubs, nothing can be tested until everything is finished.
A stub is deliberately trivial — it does no real work. Its only job is to be callable and to return something predictable, so that any fault found belongs to the module under test rather than to its dependencies.
The counterpart worth knowing for the comparison: a **driver** is the opposite, a dummy *caller* used to test a module from below when the program that will use it does not exist yet. **Mark scheme**

- One mark for method – two marks for the description
- MP1 Stub testing

Solution 5

- MP2 The modules Test_A() and Test_B()are replaced by dummy
- modules
- MP3 ... which return a known result // An output statement is displayed
- when called (to check it works) // gives expected output
- 3
- 9618/23
- October/November
- 2024
- 6
- Loop Solution

Solution 5

- Example solution:
- FUNCTION AdjustClock(ThisYear : INTEGER) RETURNS INTEGER
- DECLARE ThisDayNumber, SundayCount : INTEGER
- DECLARE ThisDate : DATE
- ThisDayNumber <- 0
- SundayCount <- 0
- REPEAT
- ThisDayNumber <- ThisDayNumber + 1
- ThisDate <- SETDATE(ThisDayNumber, 3, ThisYear)
- IF DAYINDEX(ThisDate) = 1 THEN

Solution 5

- SundayCount <- SundayCount + 1
- ENDIF
- UNTIL SundayCount = 3
- RETURN ThisDayNumber
- ENDFUNCTION
- Mark as follows:
- MP1 Function heading, parameter, ending and return type
- MP2 Declare local integer variable that is used to create a date
- MP3 Loop until 3rd Sunday found
- MP4 Attempt to use both SETDATE() and DAYINDEX() in a loop

Solution 5

- MP5 Correctly generate value of type DATE using SETDATE() in a loop
- MP6 Test if value represents a Sunday using DAYINDEX() in a loop
- MP7 Increment Sunday count in a loop and initialised correctly before loop
- MP8 Return day number of third Sunday
- Max 7 marks
- Non-loop solution
- Example solution:
- FUNCTION AdjustClock(ThisYear : INTEGER) RETURNS INTEGER
- DECLARE FirstDayIndex: INTEGER
- DECLARE ThisDate : DATE

Solution 5

- ThisDate <- SETDATE(1, 3, ThisYear)
- FirstDayIndex <- DAYINDEX(ThisDate)
- IF FirstDayIndex = 1 THEN
- ThirdSunday <- FirstDayIndex + 14 //First day is
- Sunday
- ELSE
- ThirdSunday <- 23 – FirstDayIndex // Other days
- ENDIF
- RETURN ThirdSunday
- ENDFUNCTION

Solution 5

- 7
- 9618/23
- October/November
- 2024
- 6
- Mark as follows:
- MP1 Function heading, parameter, ending and return type
- MP2 Declare local integer variable that is used to hold a day number
- MP4 Attempt to use both SETDATE() and DAYINDEX()
- MP5 Correctly generate value of type DATE using SETDATE() for first of

Solution 5

- March / specific day in March
- MP6 Test if date represents a Sunday / specific day using DAYINDEX()
- and calculate third Sunday // Correctly calculate third Sunday for a
- value returned by DAYINDEX()for one day
- MP7 Calculate third Sunday for other six days
- MP8 Return day number of third Sunday
- Max 7 marks