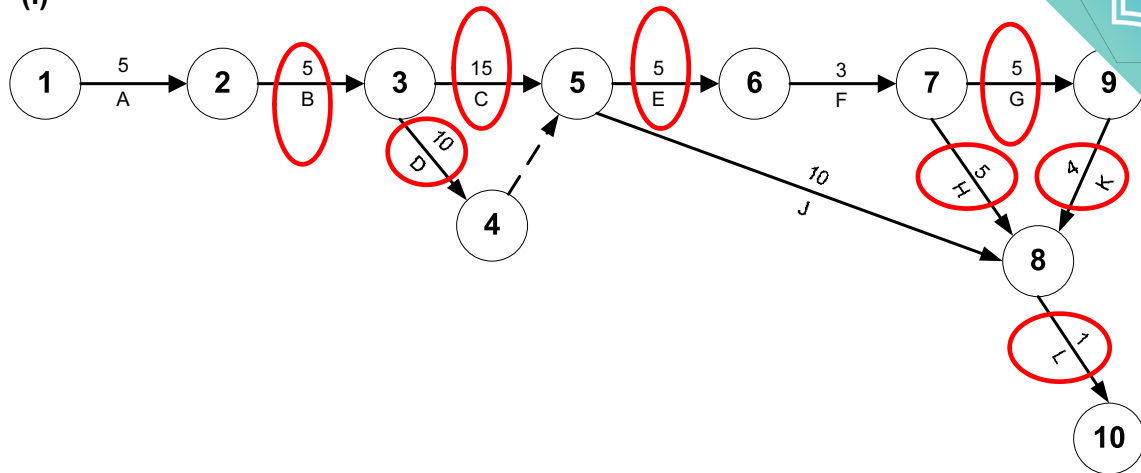


QUESTION 1.



1 (a) (i)



[max. 7]

(ii) 1 – 2 – 3 – 5 – 6 – 7 – 9 – 8 – 10

1–5 scores 1

6–10 scores 1

[2]

(iii) 43 weeks

[1]

(b) (i) week number 25

[1]

(ii) week number 32

[1]

(c) To see what activities can be done in parallel // show dependencies
To record changes to project timings

[max. 1]

Activity

1 mark for each square

Allow follow through

[1]



(b) (i)

Activity	Description	Weeks to complete
A	Write requirement specification	1
B	Produce program design	1
C	Write module code	7
D	Module testing	2
E	Integration testing	2
F	Alpha testing	2
G	Install software and carry out acceptance testing	2
H	Research and order hardware	1
J	Install delivered hardware	3
K	Write technical documentation	4
L	Write user training guide	2
M	Train users on installed hardware and software	1
N	Sign off final system	1

Activity	Week Number	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
A		■																												
B			■																											
C				■	■	■	■	■	■	■				■	■															
D										■	■			■	■															
E											■	■					■	■												
F																			■	■										
G																								■	■					
H				■																										
J																					■	■	■							
K																	■	■	■	■										
L																								■	■					
M																										■				
N																											■			

1 mark per activity (but 1 mark for activity M and N)

Notes:

C must be after E (1 or 2 later is ok)

D, E, F correct relative to C

J must start in week 20 (allow 21, 22)

G must come after the end of J (f.t.)

K finishes after or at same time as F

L finishes at the same time as G **and after the end of J** (or 1-2 weeks later)

M starts **when everything else has finished**. N after or at same time as M

[9]

(ii) week number: 26

Allow f.t.

[1]

9608/42
QUESTION 3.

Question	Answer	Marks
2(a)	1 mark for each correct line, duration and activity. <pre> graph LR 1((1)) -- "1 A" --> 2((2)) 2 -- "2 B" --> 3((3)) 3 -.-> 4((4)) 3 -- "1 D" --> 5((5)) 4 -- "3" --> 5 5 -- "12 E" --> 6((6)) 6 -- "2 F" --> 7((7)) 7 -- "4 G" --> 9((9)) 8((8)) -- "1 H" --> 9 9 -- "1 I" --> 10((10)) </pre>	7
2(b)	Dummy activity	1

Question	Answer	Marks																																																																																																																																																																																																																																																																																																																																			
5(a)	1 mark per bullet • C, D and E start at same point after B • F follows C • G follows D and H follows F • I follows G and J follows H • K follows J <table><tr><td>A</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>B</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>C</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>D</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>E</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>F</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>G</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>H</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>I</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>J</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>K</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Week number</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td></tr></table>	A																											B																											C																											D																											E																											F																											G																											H																											I																											J																											K																											Week number	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	5
A																																																																																																																																																																																																																																																																																																																																					
B																																																																																																																																																																																																																																																																																																																																					
C																																																																																																																																																																																																																																																																																																																																					
D																																																																																																																																																																																																																																																																																																																																					
E																																																																																																																																																																																																																																																																																																																																					
F																																																																																																																																																																																																																																																																																																																																					
G																																																																																																																																																																																																																																																																																																																																					
H																																																																																																																																																																																																																																																																																																																																					
I																																																																																																																																																																																																																																																																																																																																					
J																																																																																																																																																																																																																																																																																																																																					
K																																																																																																																																																																																																																																																																																																																																					
Week number	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																																																																																																																																																																																																																																																																																																												
5(b)	1 mark per bullet point to max 2: Example: • Teams can work on simultaneous/concurrent/parallel tasks • Any example of two teams working simultaneously e.g. From step 3, each team can work on a different task in week 4, 5 and 6 // tasks C, D, E can be split between different teams • Any example of working on same activities //e.g. all teams can work together on 1–2 / A, 2–3/B • Any example of working on dependent activities // e.g. if all teams work on B, then they can split up for CDE	2																																																																																																																																																																																																																																																																																																																																			
5(c)(i)	A,B,E,H,J,K	1																																																																																																																																																																																																																																																																																																																																			

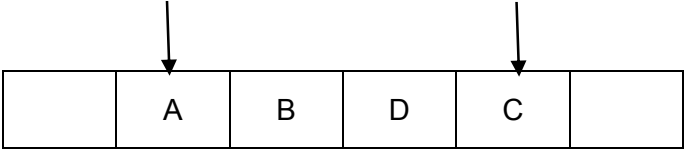
PUBLISHED

Question	Answer	Marks
5(c)(ii)	1 mark per bullet point to max 2: Example: • If there is any delay to a task which is part of the critical path • ... then the overall project will be delayed • Gives the earliest possible completion time • ... this allows you to organise/allocate resources (efficiently) • Frequently recalculate the critical path ... • ... to see if there are any delays/new critical path has arisen • Can identify where there is slack in activities • ... so they can start later without affecting the critical path	2

Question	Answer	Marks
6(a)(i)	1 mark per bullet point: • Declaring a record type // class etc. • Declaring Country as a string • Declaring Pointer as an integer Example: <pre> TYPE ListElement DECLARE Country : STRING DECLARE Pointer : INTEGER ENDTYPE </pre>	3

Question	Answer	Marks
1(a)(i)	1 mark for each correct Activity and time • B 3 • C 10 (following B) • D 7 (following B) • E 3 (following C and D) • G 2 (following F) • H 2 (following F)	6
1(a)(ii)	The shortest time to complete the project // the sequence of activities that must be completed to avoid delaying the project	1
1(a)(iii)	1 mark for identify, max 1 for description • GANTT • A table that has time across the top and activities on the left, boxes are coloured to show dependencies and find critical path // colour in the boxes to show the length of time for each task	2

PUBLISHED

Question	Answer	Marks
1(b)(i)	A, B, C and D in correct places with no alteration to start and end pointer Start Pointer End Pointer 	1
1(b)(ii)	1 mark per bullet point • correct jobs in correct order... • ... correct location of start pointer • ... correction location of new end pointer End Pointer Start Pointer 	3
1(b)(iii)	1 mark from: • An error message would be generated	1

Question	Answer	Marks
1(b)(iv)	1 mark for each correct line <pre> FUNCTION Remove RETURNS STRING DECLARE PrintJob : STRING IF StartPointer = EndPointer THEN RETURN "Empty" ELSE PrintJob ← Queue[StartPointer] IF StartPointer = 5 THEN StartPointer ← 0 ELSE StartPointer ← StartPointer + 1 ENDIF RETURN PrintJob ENDIF ENDFUNCTION </pre>	4
1(b)(v)	1 mark per bullet point • A stack is Last In First Out (LIFO) while a queue is First In First Out (FIFO) • The queue removes and returns the element at start pointer // item is removed from the start/head // • A stack would remove and return the element at end pointer // item is removed from the end	2

Question	Answer	Marks
1(c)	1 mark for each correct transition with arrow <pre>graph TD; Start((Start)) --> Idle((Printer idle)); Idle -- "Print job sent" --> Active((Printer active)); Active -- "Timeout" --> Idle; Active -- "Print job added to queue" --> Received((Print job received)); Received -- "Print job in progress" --> Successful((Print job successful)); Received -- "Print job in progress" --> Unsuccessful((Job unsuccessful)); Successful -- "Check print queue" --> Active; Unsuccessful -- "Error message" --> Active;</pre>	5

Question	Answer	Marks																																																																											
1(d)(i)	1 mark per bullet - Way of modelling logic - Show all possible outputs // shows every possible outcome // all outcomes based on the inputs - Determine which action to take in specific conditions // how different conditions affect the actions/outcomes	2																																																																											
1(d)(ii)	1 mark for each row Accept Y/X/ticks as long as clear which are Y. Accept N/X/– for empty spaces <table><tr><td></td><td></td><td colspan="8">Rules</td></tr><tr><td rowspan="3">Conditions</td><td>Document printed, but quality is poor</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td><td>N</td><td>N</td><td>N</td><td>N</td></tr><tr><td>Error light is flashing on printer</td><td>Y</td><td>Y</td><td>N</td><td>N</td><td>Y</td><td>Y</td><td>N</td><td>N</td></tr><tr><td>Document printed, but paper size is incorrect</td><td>Y</td><td>N</td><td>Y</td><td>N</td><td>Y</td><td>N</td><td>Y</td><td>N</td></tr><tr><td rowspan="4">Actions</td><td>Check connection from computer to printer</td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td></tr><tr><td>Check ink status</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>Check if there is a paper jam</td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td></tr><tr><td>Check paper size selected</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td></tr></table>			Rules								Conditions	Document printed, but quality is poor	Y	Y	Y	Y	N	N	N	N	Error light is flashing on printer	Y	Y	N	N	Y	Y	N	N	Document printed, but paper size is incorrect	Y	N	Y	N	Y	N	Y	N	Actions	Check connection from computer to printer						X			Check ink status	X	X	X	X					Check if there is a paper jam						X			Check paper size selected	X		X		X		X		4
		Rules																																																																											
Conditions	Document printed, but quality is poor	Y	Y	Y	Y	N	N	N	N																																																																				
	Error light is flashing on printer	Y	Y	N	N	Y	Y	N	N																																																																				
	Document printed, but paper size is incorrect	Y	N	Y	N	Y	N	Y	N																																																																				
Actions	Check connection from computer to printer						X																																																																						
	Check ink status	X	X	X	X																																																																								
	Check if there is a paper jam						X																																																																						
	Check paper size selected	X		X		X		X																																																																					

Question	Answer										Marks
1(d)(iii)	1 mark each for each correct column										5
	Accept –/X for empty spaces. Accept Y/X/Ticks as long as clear which are used										
			Rules								
	Conditions	Document printed, but quality is poor	Y	Y	N	N	N				
		Error light is flashing on printer				Y	N				
		Document printed, but paper size is incorrect	Y	N	Y	N	N				
	Actions	Check connection from computer to printer				X					
		Check ink status	X	X							
		Check if there is a paper jam				X					
		Check paper size selected	X		X						

PUBLISHED

Question	Answer	Marks
1(e)(i)	1 mark per bullet point to max 4 • Method header and close (where necessary) with three parameters • Initialised PrintID, FirstName, LastName and Credits ... • ... to the parameters • Initialised Credits to 50 PYTHON <pre>def __init__(self, NewFN, NewLN, NewPrintID): self.__PrintID = NewPrintID self.__FirstName = NewFN self.__LastName = NewLN self.__Credits = 50</pre> PASCAL <pre>Constructor NewPrintAccount.Create(NewFN, NewLN, NewPrintID); begin PrintID := NewPrintID; FirstName = NewFN; LastName = NewLN; Credits := 50; end;</pre> VB <pre>Public Sub New(NewFN, NewLN, NewPrintID As String) PrintID = NewPrintID FirstName = NewFN LastName = NewLN Credits = 50 End Sub</pre>	4

Question	Answer	Marks
1(e)(ii)	1 mark per bullet point • method/procedure header (and close where appropriate) taking a parameter • <code>FirstName</code> is set to parameter PYTHON <pre>def __SetFirstName(self, NewFirstName): self.__FirstName = NewFirstName</pre> PASCAL <pre>procedure SetFirstName(newFirstName : String); begin FirstName := newFirstName; end;</pre> VB <pre>public sub SetFirstName(NewFirstName As String) FirstName = NewFirstName End Sub</pre>	2

PUBLISHED

Question	Answer	Marks
1(e)(iii)	1 mark per bullet point • concatenates <code>FirstName</code>, space and <code>LastName</code> ... • ... function/method header without parameter and returns (generated) value PYTHON <pre>def __GetName(self): return(self.__FirstName + " " + self.__LastName)</pre> PASCAL <pre>function GetName(); begin result := FirstName + " " + LastName end;</pre> VB <pre>public function GetName() As String return(FirstName & " " & LastName) End Function</pre>	2

PUBLISHED

Question	Answer	Marks
1(e)(iv)	1 mark per each correct bullet point from: • Procedure/method header and close (where necessary) passing MoneyInput • At least 3 constants (e.g. freecredit10, freecredit20, twenty, 10, creditperdollar) • If MoneyInput < 10 calculate MoneyInput * 25 • If MoneyInput > 9 and MoneyInput < 20 then calculate MoneyInput * 25 + 25 • If MoneyInput > 19 then calculate MoneyInput * 25 + 50 • All three correct calculations add to Credits, not overwrite • Efficient IF (i.e. elseif) PYTHON <pre>def __AddCredits(self, MoneyInput): CreditPerDollar = 25 FreeCredit10 = 25 FreeCredit20 = 50 Twenty = 20 Ten = 10 if MoneyInput >= Twenty: Credits = Credits + (MoneyInput * CreditPerDollar) + FreeCredit20 elif MoneyInput >= Ten: Credits = Credits + (MoneyInput * CreditPerDollar) + FreeCredit10 else: Credits = Credits + (MoneyInput * CreditPerDollar)</pre>	6

Question	Answer	Marks
1(e)(iv)	PASCAL <pre> procedure AddCredits(MoneyInput : Real); const CreditPerDollar = 25 const FreeCredit10 = 25 const FreeCredit20 = 50 const Twenty = 20 const Ten = 10 begin If MoneyInput >= Twenty Then Credits := Credits + (MoneyInput * CreditPerDollar) + FreeCredit20; Else If MoneyInput >= Ten Then Credits := Credits + (MoneyInput * CreditPerDollar) + FreeCredit10; Else Credits := Credits + (MoneyInput * CreditPerDollar); end;</pre> VB.NET <pre> Public Sub AddCredits(MoneyInput As Integer) Const CreditPerDollar As Integer = 25 Const FreeCredit10 As Integer = 25 Const FreeCredit20 AS integer = 50 Const Twenty As Integer = 20 Const Ten AS Integer = 10 If MoneyInput >= Twenty Then Credits = Credits + (MoneyInput * CreditPerDollar) + FreeCredit20 Else If MoneyInput >= Ten Then Credits = Credits + (MoneyInput * CreditPerDollar) + FreeCredit10 Else Credits = Credits + (MoneyInput * CreditPerDollar) End If End Sub</pre>	

Question	Answer	Marks
1(e)(v)	1 mark per bullet • Declaring <code>StudentAccounts</code> as array of 1000 elements... • ...of type <code>PrintAccount</code> <code>DECLARE StudentAccounts ARRAY[0:999] OF PrintAccount</code>	2

PUBLISHED

Question	Answer	Marks
1(e)(vi)	1 mark per bullet point to max 8 • Generating ID with '1' at the end all in lowercase from parameters • Loop through array to last occupied element ... • ... Check if the <code>PrintID</code> already exists • ... using <code>GetPrintID()</code> • ... increment number at end of <code>PrintID</code> • Create a new instance of <code>PrintAccount</code> ... • ... sending <code>FirstName</code>, <code>LastName</code>, <code>PrintID</code> as parameters • adding new account to <code>StudentAccounts</code> at position <code>NumberStudents</code> • Increment <code>NumberStudents</code> VB.NET <pre> Sub CreateId(firstName, lastName) Dim count As Integer Dim PrintID = Left(firstName, 3).ToLower & Left(lastname, 3).ToLower & "1" Dim studentAdd As Integer = 0 If numberStudents <> 0 Then For x = 0 To numberStudents - 1 If studentAccounts(x).getPrintID() = username Then PrintID = PrintID + 1 username = Left(firstName, 3).ToLower & Left(lastname, 3).ToLower & PrintID.ToString End If Next studentAdd = numberStudents End If studentAccounts(studentAdd) = New printAccount(firstName, lastname, username) numberStudents = numberStudents + 1 </pre>	8

Question	Answer	Marks
1(e)(vi)	Python <pre>def CreateID(firstname, lastname): count = 0 PrintID = firstname[0:3].lower() + lastname[-3].lower() + "1" StudentAdd = 0 if numberStudents != 0: for x in range(0, numberStudents): if studentAccounts[x].getPrintID() == username: PrintID = PrintID + 1 username = firstname[0:3].lower() + lastname[0:3].lower + str(PrintID) studentAdd = numberStudents studentAccounts[studentAdd] = printAccount(firstname, lastname, username) numberStudents = numberStudents + 1</pre> Pascal <pre>procedure CreateID(firstname : String, lastname: String); var count : Integer; studentAdd : Integer; PrintID : String; begin studentAdd := 0; PrintID := LowerCase(substr(firstname,0,3)) + LowerCase(substr(lastname,0,3))+ "1"; if numberStudents <> 0: for x := 0 To numberStudents - 1; if studentAccounts[x].getPrintID() = username: PrintID := PrintID + 1; username := LowerCase(substr(firstname, 3) + LowerCase(substr(lastname,0,3))+str(PrintID); studentAdd := numberStudents; studentAccounts[studentAdd] := printAccount.Create(firstname, lastname, username); numberStudents := numberStudents + 1</pre>	

Question	Answer	Marks
1(a)(i)	<pre> graph LR 1((1)) -- "A 1" --> 2((2)) 2 -- "B 1" --> 3((3)) 3 -- "C 3" --> 4((4)) 4 -- "D 1" --> 5((5)) 4 -- "E 7" --> 6((6)) 5 -- "F 2" --> 7((7)) 5 -- "G 3" --> 8((8)) 6 -.-> 7 8 -.-> 7 7 -- "H 1" --> 9((9)) 9 -- "I 2" --> 10((10)) 9 -- "J 2" --> 11((11)) 10 -.-> 11 </pre> 1 mark each bullet point: • B 1 • C 3 (following B) • D 10 (following C) • G 3 (following D) and nothing on dummy • E 7 (following C) and nothing on dummy • H 1 in position • J 2 (following H) and nothing on dummy	7
1(a)(ii)	1 mark: • The next activity is dependent on the previous but there is no activity	1

Question	Answer										Marks
1(b)	1 mark per row										3
			Rules								
Conditions	Public Holiday		Y	Y	Y	Y	N	N	N	N	
	Hours ≥ 160		Y	Y	N	N	Y	Y	N	N	
	Pension		Y	N	Y	N	Y	N	Y	N	
Actions	3% bonus payment		X	X	X	X					
	5% bonus payment		X	X			X	X			
	4% Pension payment		X		X		X		X		

PUBLISHED

Question	Answer		Marks
1(c)(i)	1 mark per bullet point • inheritance (arrow filled or unfilled) • constructor and SetHoursThisWeek method for ApprenticeshipEmployee • HourlyRate and HoursThisWeek attributes for ApprenticeshipEmployee	Employee EmployeeID : STRING Name : STRING Address : STRING DateOfBirth : DATE Constructor () GetEmployeeID() GetName() GetAddress() GetDateOfBirth() SetEmployeeID() SetName() SetAddress() SetDateOfBirth() SalaryEmployee MonthlyPayment : CURRENCY HoursThisMonth : REAL PublicHoliday : BOOLEAN Pension : BOOLEAN Constructor() GetMonthlyPayment() GetPension() GetPublicHoliday() GetHoursThisMonth() SetMonthlyPayment() SetPension() SetPublicHoliday() SetHoursThisMonth() ApprenticeshipEmployee HourlyRate : CURRENCY/REAL HoursThisWeek : REAL/INTEGER Constructor () SetHoursThisWeek () GetHourlyRate() GetHoursThisWeek() SetHourlyRate()	3

PUBLISHED

Question	Answer	Marks
1(c)(ii)	1 mark per bullet point • Constructor header and close (where necessary) • All 4 values sent as parameters (ID, Name, Address, DateOfBirth) with any • Attributes/properties set to a value ... • ... that are the parameters Example code: Pascal <pre>Constructor Employee.init(ID, NewName, NewAddress, NewDateOfBirth : String); begin EmployeeID := ID; Name := NewName; Address := NewAddress; DateOfBirth := NewDateOfBirth; end;</pre> Python <pre>def __init__(self, ID, NewName, NewAddress, NewDateOfBirth): self.__EmployeeID = ID self.__Name = NewName self.__Address = NewAddress self.__DateOfBirth = NewDateOfBirth</pre> VB.NET <pre>Public Sub New(ID, NewName, NewAddress, NewDateOfBirth) EmployeeID = ID Name = NewName Address = NewAddress DateOfBirth = NewDateOfBirth End Sub</pre>	4

PUBLISHED

Question	Answer	Marks
1(c)(iii)	1 mark per bullet point • Get method header and close (where needed) with no parameters • Returns the attribute/property <code>EmployeeID</code> Example code: Pascal <pre>Function Employee.GetEmployeeID() : String; result := EmployeeID; end;</pre> Python <pre>def GetEmployeeID(self): return self.__EmployeeID</pre> VB.NET <pre>Public Function GetEmployeeID() As String Return EmployeeID End Function</pre>	2

PUBLISHED

Question	Answer	Marks
1(c)(iv)	1 mark per bullet point • Set method/procedure header and close (where needed) with parameter • Sets <code>EmployeeID</code> to value of parameter Example code: Pascal <pre>procedure Employee.SetEmployeeID(NewID: String); EmployeeID := NewID end;</pre> Python <pre>def SetEmployeeID(self, NewID): self.__EmployeeID = NewID</pre> VB.NET <pre>Public Sub SetEmployeeID(NewID) EmployeeID = NewID End Sub</pre>	2

PUBLISHED

Question	Answer	Marks
1(c)(v)	1 mark per bullet point • Set method/function header and close (where needed) with parameter • Checking value of parameter for both true and false ... • ... If valid – setting Pension to parameter and returning True • ... If not valid – not setting Pension and returning False Example code: Pascal <pre>Function salaryEmployee.SetPension(NewPension) : boolean; IF NewPension = true or NewPension = false THEN Pension := NewPension; Result := true; ELSE Result := false; end;</pre> Python <pre>def SetPension(self, NewPension): if NewPension == True Or NewPension == False: self.__Pension = NewPension return True else: return False</pre> VB.NET <pre>Public Function SetPension(NewPension) AS Boolean If NewPension = True Or NewPension = False Then Pension = NewPension Return True Else Return False End If End Function</pre>	4

PUBLISHED

Question	Answer	Marks
1(c)(vi)	1 mark per bullet point to max 8 • Function header and close (where needed) with at least one parameter • Constants used for hours bonus, month bonus, holiday bonus, pension cost(at least 3) • Checking if Hours is ≥ 160, calculating bonus payment ($\text{monthlypay} * 0.05$) • Checking if pension is true, calculation pension to pay ($\text{monthlypay} * 0.04$) • Checking if public holiday is true, calculation of bonus payment ($\text{monthlypay} * 0.03$) • ... all 3 Hours, Pension and <code>PublicHoliday</code> accessed from the parameter using Get methods • Adding both bonus payments to basic salary and deducting pension from salary • ... basic salary accessed by using <code>GetMonthlyPayment</code> with the parameter • Outputting the total final bonus and pension with appropriate message • Returning the new salary Example code: Pascal <pre>Function CalculateMonthlySalary(TheEmployee : SalaryEmployee) : real; var BonusPayment : real; PensionPayment : real; BasicSalary : real; const HoursBonus : real = 0.05; HoursMonthBonus : integer = 160; PensionCost : real = 0.04; PublicHolidayBonus : real = 0.03; begin BonusPayment :=0; PensionPayment :=0; BasicSalary :=0;</pre>	8

PUBLISHED

Question	Answer	Marks
1(c)(vi)	<pre> BasicSalary := TheEmployee.GetMonthlyPayment(); IF TheEmployee.GetHoursThisMonth() >= HoursMonthBonus THEN BonusPayment := BasicSalary * HoursBonus; IF TheEmployee.GetPension() = True THEN PensionPayment := BasicSalary * PensionCost; IF TheEmployee.GetPublicHoliday() = True THEN BonusPayment := BonusPayment + BasicSalary * PublicHolidayBonus; writeln("The pension payment is " & PensionPayment); writeln ("The bonus payment is " & BonusPayment); MonthlySalary := BasicSalary + BonusPayment - PensionPayment; result := MonthlySalary; end; Python def CalculateMonthlySalary(TheEmployee): BonusPayment = 0 PensionPayment = 0 HoursBonus = 0.05 HoursMonthBonus = 160 PensionCost = 0.04 PublicHolidayBonus = 0.03 BasicSalary = TheEmployee.GetMonthlyPayment() if TheEmployee.GetHoursThisMonth() >= HoursMonthBonus: BonusPayment = BasicSalary * HoursBonus </pre>	

Question	Answer	Marks
1(c)(vi)	<pre> if TheEmployee.GetPension() == true: PensionPayment = BasicSalary * PensionCost if TheEmployee.GetPublicHoliday() == true: BonusPayment = BonusPayment + BasicSalary * PublicHolidayBonus print("The pension payment is ", str(PensionPayment)) print("The bonus payment is ", str(BonusPayment)) MonthlySalary = BasicSalary + BonusPayment - PensionPayment return MonthlySalary </pre> VB.NET Public Function CalculateMonthlySalary(TheEmployee As SalaryEmployee) As Double Dim BonusPayment As Single = 0 Dim PensionPayment As Single = 0 Dim MonthlySalary As Single = 0 Const HoursBonus As Single = 0.05 Const HoursMonthBonus As Integer = 160 Const PensionCost As Single = 0.04 Const PublicHolidayBonus As Single = 0.03 Dim BasicSalary As Double = TheEmployee.GetMonthlyPayment() If TheEmployee.GetHoursThisMonth() >= HoursMonthBonus Then BonusPayment = BasicSalary * HoursBonus End If If TheEmployee.GetPension() = True Then PensionPayment = BasicSalary * PensionCost End If If TheEmployee.GetPublicHoliday() = True Then BonusPayment = BonusPayment + BasicSalary * PublicHolidayBonus End If	

Question	Answer	Marks
1(c)(vi)	<pre>Console.WriteLine("The pension payment is " & PensionPayment) Console.WriteLine("The bonus payment is " & BonusPayment) MonthlySalary = BasicSalary + BonusPayment - PensionPayment Return MonthlySalary End Function</pre>	
1(d)	Polymorphism	1