

QUESTION 6.



- 3 A shop gives some customers a discount on goods totalling more than \$20. The discounts are:
- 5% for goods totalling more than \$100
 - 5% with a discount card
 - 10% with a discount card and goods totalling more than \$100

(a) Complete the decision table.

Conditions	goods totalling more than \$20	Y	Y	Y	Y	N	N	N	N
	goods totalling more than \$100	Y	Y	N	N	Y	Y	N	N
	have discount card	Y	N	Y	N	Y	N	Y	N
Actions	No discount								
	5% discount								
	10% discount								

[4]

(b) Simplify your solution by removing redundancies.

Conditions	goods totalling more than \$20								
	goods totalling more than \$100								
	have discount card								
Actions	No discount								
	5% discount								
	10% discount								

[5]

- 



QUESTION 7.



3 An insurance company calculates the cost of car insurance from a basic price.

The driver may:

- get a discount on the basic price of the insurance
- have to pay an extra charge

The decision is arrived at as follows:

- for a driver aged 25 or over:
 - 5% discount if no previous accident
 - no discount if a previous accident
- for a driver under the age of 25:
 - 5% discount if no previous accident and licence held for 3 or more years
 - no discount if a previous accident but licence held for 3 or more years
 - no discount if no previous accident but licence held for less than 3 years
 - 10% extra charge if a previous accident and licence held for less than 3 years

(a) Complete the decision table.

Conditions	Age under 25	Y	Y	Y	Y	N	N	N	N
	Previous accident	Y	Y	N	N	Y	Y	N	N
	Licence held for 3 or more years	Y	N	Y	N	Y	N	Y	N
Actions	10% extra charge								
	No discount								
	5% discount								

[6]

(b) Simplify your solution by removing redundancies.

Conditions	Age under 25								
	Previous accident								
	Licence held for 3 or more years								
Actions	10% extra charge								
	No discount								
	5% discount								

[3]

- 



QUESTION 8.



1 Students are choosing their A Level subjects based on their IGCSE subject results.

A student can take:

- Computer Science, if they have a grade C in Maths or a grade C in Computer Science
- Maths, if they have a grade C in Maths
- Physics, if they have a grade C in Science and a grade C in Maths.

(a) Complete the decision table.

		Column							
		1	2	3	4	5	6	7	8
Conditions	Grade C in Computer Science	Y	Y	Y	Y	N	N	N	N
	Grade C in Maths	Y	Y	N	N	Y	Y	N	N
	Grade C in Science	Y	N	Y	N	Y	N	Y	N
Actions	Take Computer Science								
	Take Maths								
	Take Physics								

[4]

(b) Simplify your solution by removing redundancies.

		Column							
		S	T	U	V	W	X	Y	Z
Conditions	Grade C in Computer Science								
	Grade C in Maths								
	Grade C in Science								
Actions	Take Computer Science								
	Take Maths								
	Take Physics								

[3]



(c) Show how the columns from **part (a)** were simplified to create the columns in **part (b)**.

For example, if columns 5, 6 and 7 were simplified to create column X, then you would write X = 5 + 6 + 7 in your answer.

.....

.....

.....

.....

.....

.....[3]

QUESTION 9.



- 1 Each student at CIE University needs a printing account to print documents on their computers.

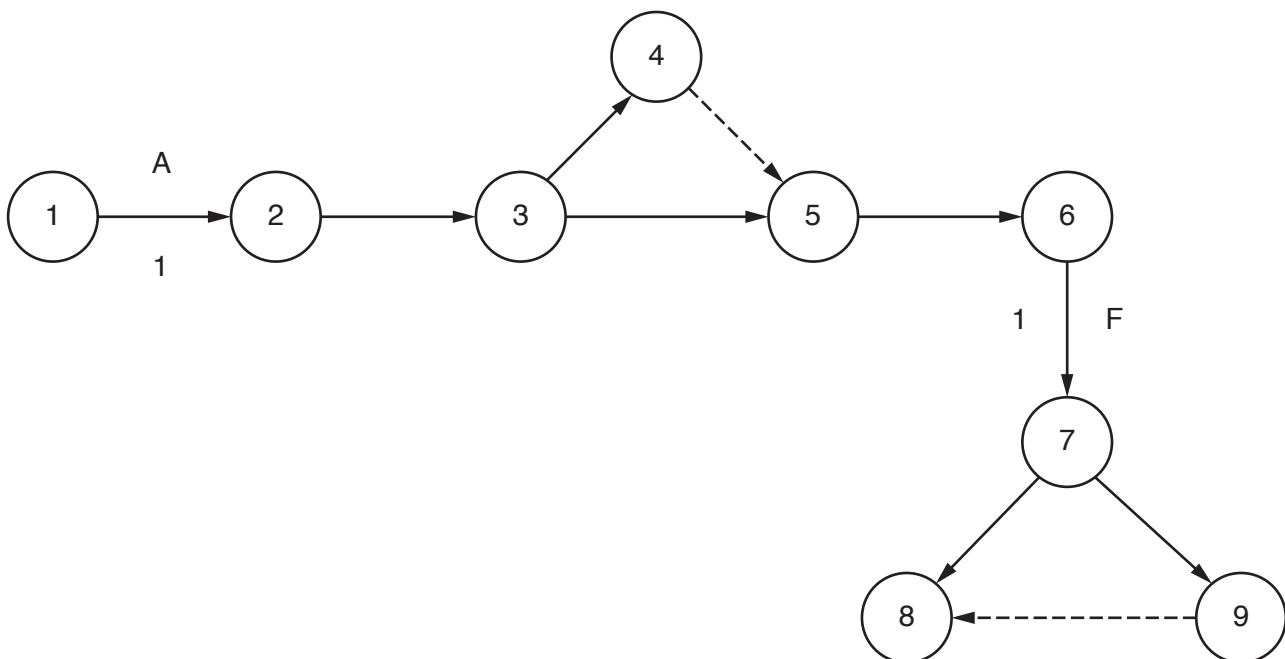
The university is developing software to manage each student's printing account and the process.

(a) Developing the software will include the following activities.

Activity	Description	Time in weeks	Predecessor
A	Identify requirements	1	-
B	Produce design	3	A
C	Write code	10	B
D	Test modules	7	B
E	Final system black-box testing	3	C, D
F	Install software	1	E
G	Acceptance testing	2	F
H	Create user documentation	2	F

- (i) Add the correct activities and times to the following Program Evaluation Review Technique (PERT) chart for the software development.

Two activities and times have been done for you.





- (ii) State what is meant by the **critical path** in a PERT chart.

.....

.....

- (iii) Identify **and** describe a project planning technique, other than a PERT chart.

.....

.....

.....

..... [2]

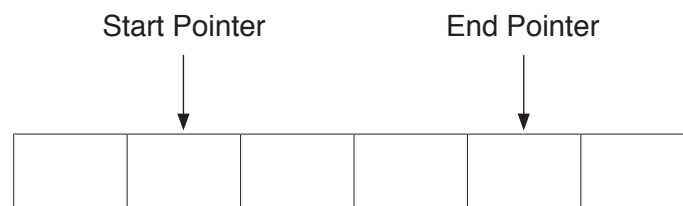
- (b) When a student prints a document, a print job is created. The print job is sent to a print server.

The print server uses a queue to hold each print job waiting to be printed.

- (i) The queue is circular and has six spaces to hold jobs.

The queue currently holds four jobs waiting to be printed. The jobs have arrived in the order A, B, D, C.

Complete the diagram to show the current contents of the queue.



[1]

- (ii) Print jobs A and B are now complete. Four more print jobs have arrived in the order E, F, G, H.

Complete the diagram to show the current contents and pointers for the queue.



[3]

- (iii) State what would happen if another print job is added to the queue in the status in **part (b)(ii)**.

.....

..... [1]



- (iv) The queue is stored as an array, `Queue`, with six elements. The function `Remove` removes a print job from the queue and returns it.

Complete the following **pseudocode** for the function `Remove`.

```

FUNCTION Remove RETURNS STRING
    DECLARE PrintJob : STRING
    IF ..... = EndPointer
        THEN
            RETURN "Empty"
        ELSE
            PrintJob ← Queue[.....]
            IF StartPointer = .....
                THEN
                    StartPointer ← .....
                ELSE
                    StartPointer ← StartPointer + 1
            ENDIF
            RETURN PrintJob
        ENDIF
    ENDFUNCTION
  
```

[4]

- (v) Explain why the circular queue could not be implemented as a stack.

.....

.....

.....

..... [2]

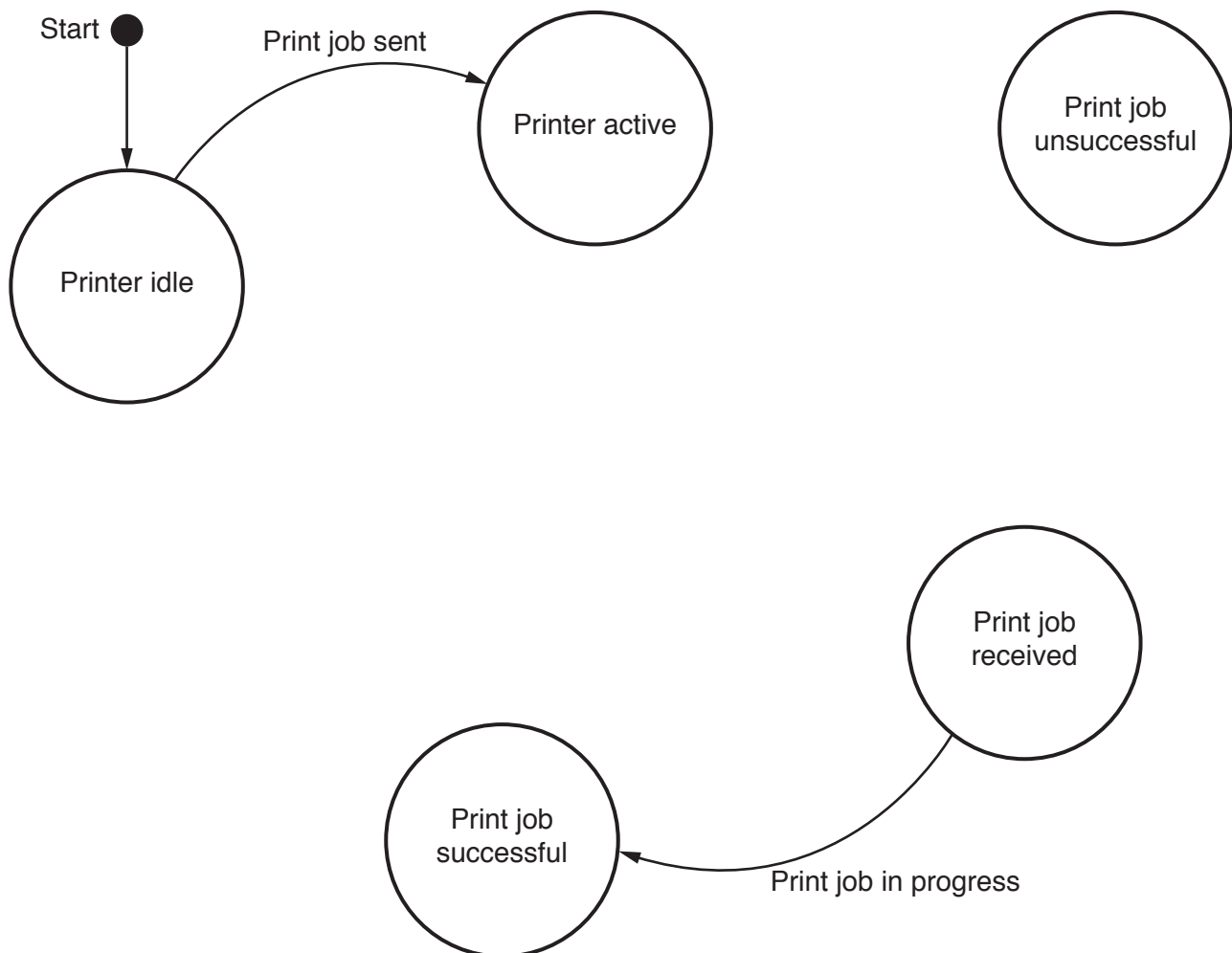


(c) The university wants to analyse how a printer and a print server deal with the

The following table shows the transitions from one state to another for the process

Current state	Event	Next state
Printer idle	Print job sent	Printer active
Printer active	Print job added to queue	Print job received
Print job received	Print job in progress	Print job successful
Print job received	Print job in progress	Print job unsuccessful
Print job successful	Check print queue	Printer active
Print job unsuccessful	Error message displayed	Printer active
Printer active	Timeout	Printer idle

Complete the state-transition diagram for the table.





- (d) The university wants to assess troubleshooting issues with a printer. It wants a decision table to do this.

The troubleshooting actions are:

- check the connection from computer to printer, if the error light is flashing **and** the document has not been printed
- check the ink status, if the quality is poor
- check whether there is a paper jam, if the error light is flashing **and** the document has not been printed
- check the paper size selected, if the paper size is incorrect.

- (i) Describe the purpose of a decision table.

.....

.....

.....

..... [2]

- (ii) Complete the rules for the actions in the following decision table.

		Rules							
Conditions	Document printed but the 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								
	Check ink status								
	Check if there is a paper jam								
	Check the paper size selected								

[4]

(iii) Simplify your solution by removing redundancies.



		Rules							
Conditions	Document printed but the quality is poor								
	Error light is flashing on printer								
	Document printed but paper size is incorrect								
Actions	Check connection from computer to printer								
	Check ink status								
	Check if there is a paper jam								
	Check the paper size selected								

[5]



- (e) There are 1000 students at the university. They will each require a printing account.

Students need to buy printing credits that will be added to their account. Each print job uses one printing credit.

The university needs software to keep track of the number of printing credits each student has in their account. The university has decided to implement the software using object-oriented programming (OOP).

The following diagram shows the design for the class `PrintAccount`. This includes the attributes and methods.

PrintAccount	
<code>FirstName</code>	<code>: STRING // parameter sent to Constructor()</code>
<code>LastName</code>	<code>: STRING // parameter sent to Constructor()</code>
<code>PrintID</code>	<code>: STRING // parameter sent to Constructor()</code>
<code>Credits</code>	<code>: INTEGER // initialised to 50</code>
<code>Constructor()</code>	<code>// instantiates an object of the PrintAccount class, // and assigns initial values to the attributes</code>
<code>GetName()</code>	<code>// returns FirstName and LastName concatenated // with a space between them</code>
<code>GetPrintID()</code>	<code>// returns PrintID</code>
<code>SetFirstName()</code>	<code>// sets the FirstName for a student</code>
<code>SetLastName()</code>	<code>// sets the LastName for a student</code>
<code>SetPrintID()</code>	<code>// sets the PrintID for a student</code>
<code>AddCredits()</code>	<code>// increases the number of credits for a student</code>
<code>RemoveCredits()</code>	<code>// removes credits from a student account</code>



- (i) Write **program code** for the `Constructor()` method.

Programming language

Program code

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

- (ii) Write **program code** for the `SetFirstName()` method.

Programming language

Program code

.....

.....

.....

.....

..... [2]

- (iii) Write **program code** for the `GetName()` method.

Programming language

Program code

.....

.....

.....

.....

.....

..... [2]

- 





- (v) A global array, `StudentAccounts`, stores 1000 instances of `PrintAccount`.

Write **pseudocode** to declare the array `StudentAccounts`.

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

(vi) The main program has a procedure, `CreateID()`, that:

- takes the first name and last name as parameters
- creates `PrintID` that is a concatenation of:
 - the first three letters of the first name in lower case
 - the first three letters of the last name in lower case
 - the character '1'for example, the name Bill Smith would produce "`bilsmi1`"
- checks if the `PrintID` created already exists in the global array `StudentAccounts`:
 - If `PrintID` does not exist, it creates an instance of `PrintAccount` in the next free index in `StudentAccounts`.
 - If `PrintID` does exist, the number is incremented until a unique ID is created, for example, "`bilsmi2`". It then creates an instance of `PrintAccount` in the next free index in `StudentAccounts`.

The global variable `NumberStudents` stores the number of print accounts that have currently been created.

Write **program code** for the procedure `CreateID()`. Do not write the procedure header.

Programming language

Program code

[illegible]

[illegible]

QUESTION 10.



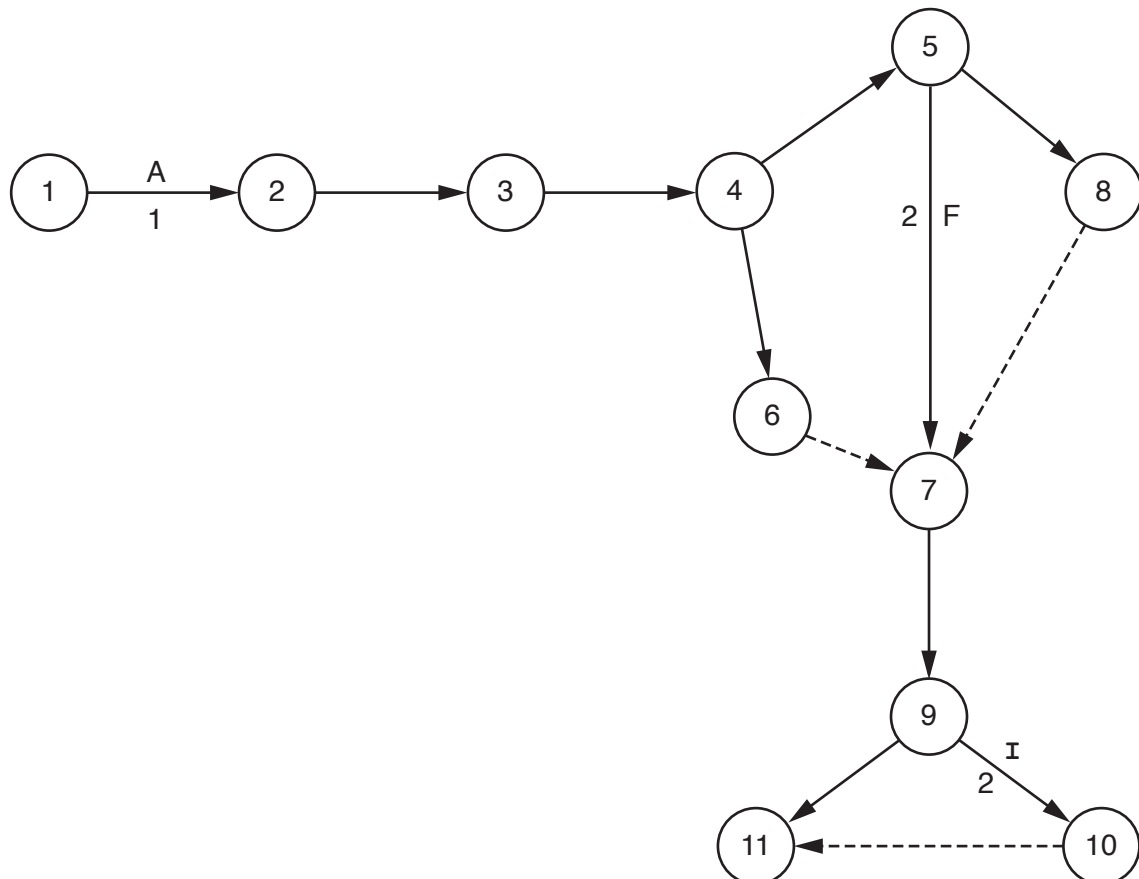
1 A technology company needs software to calculate how much each employee should be paid.

(a) Developing the software will involve the following activities:

Activity	Description	Time to complete (weeks)	Predecessor
A	Identify requirements	1	–
B	Observe current system	1	A
C	Create algorithm design	3	B
D	Write code	10	C
E	Test modules	7	C
F	White box testing	2	D
G	Black box testing	3	D
H	Install software	1	E, F, G
I	Acceptance testing	2	H
J	Create user documentation	2	H

(i) Add the correct activities and times to the following Program Evaluation Review Technique (PERT) chart for the software development.

Three of the activities and times have been done for you.





- (ii) The dashed line connecting nodes 10 and 11 indicates a dummy activity.

State the purpose of a dummy activity.

.....
 [1]

- (b) A bonus payment may be added to an employee's salary. A pension payment may also be subtracted from an employee's salary.

The company needs to assess what additions and subtractions should be made to the salary of each employee. There are three conditions to check:

- If the employee has worked a public holiday, they receive a 3% bonus payment.
- If the employee has worked 160 or more hours in a month, they receive an additional 5% bonus payment.
- If the employee pays into a pension, the company subtracts 4% for the pension payment.

Complete the decision table to show the additions and subtractions.

		Rules							
Conditions	Public holiday	Y	Y	Y	Y	N	N	N	N
	Hours $\geq$ 160	Y	Y	N	N	Y	Y	N	N
	Pension	Y	N	Y	N	Y	N	Y	N
Actions	3% bonus payment								
	5% bonus payment								
	4% pension payment								

[3]



- (c) The company decides to implement a program for the software using programming (OOP).

Each employee has a unique employee ID, name, address and date of birth. There are two types of employee: salary and apprenticeship.

Salaries employees are paid a fixed monthly payment. The hours a salary employee works in a month are recorded to calculate bonus payments. They may receive bonus payments and make pension payments (given in **part(b)**).

Apprenticeship employees are paid weekly. They receive an hourly rate of pay. Apprenticeship employees do not receive bonus payments or make pension payments.

- (i) Complete the following class diagram for the program.

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() GetHoursThisMonth() GetPublicHoliday() GetPension() SetMonthlyPayment() SetHoursThisMonth() SetPublicHoliday() SetPension()

ApprenticeshipEmployee
.....
..... GetHourlyRate() GetHoursThisWeek() SetHourlyRate()

- All properties are sent as parameters.

Program code

[4]

- The `get` method returns the value of the `EmployeeID` property.

Program code

..... [2]

The set method takes the new value as its parameter.

Program code

(v) Write **program code** for the `SetPension()` method in the `SalaryEmployee()` class.

- Programming language

[4]



Question 1 continues on the next page.

(vi) A `SalaryEmployee` is paid a fixed monthly payment.

- If the employee has worked a public holiday, they receive a 3% bonus. This is calculated from their `MonthlyPayment`.
- If the employee has worked 160 or more hours in a month, they receive an additional 5% bonus payment, calculated from their `MonthlyPayment`.
- If the employee pays into a pension, 4% will be subtracted from their `MonthlyPayment`.

Monthly salary is the final payment the employee receives.

For example, Chris is a `SalaryEmployee`. His fixed `MonthlyPayment` is \$1000. He has worked a public holiday. He has worked 165 hours this month. He pays into a pension.

- The public holiday bonus is \$30 (3% of \$1000)
- The hours worked bonus payment is \$50 (5% of \$1000)
- The pension payment is \$40 (4% of \$1000)

Chris's monthly salary is calculated as $(\$1000 + \$30 + \$50) - \$40 = \$1040$

The function `CalculateMonthlySalary()` is used to calculate the monthly salary. It:

- takes a `SalaryEmployee` as a parameter
- calculates the bonus payments and pension payment
- outputs the pension payment and total bonus payment
- calculates and returns the monthly salary.

Write **program code** for the function `CalculateMonthlySalary()`.

Programming language

Program code

[illegible]



- (d)** Noona describes an example of a feature of object-oriented programming (OOP).
“One method exists in the parent class but is overwritten in the child class, differently.”

Identify the feature Noona has described.

