Student number: _____

Teacher: Dr V. Tošić

Q01	/01
Q02	/01
Q03	/01
Q04	/01
Q05	/02
Q06	/02
Q07	/02
Q08	/02
Q09	/04
Q10	/03
Q11	/03
Q12	/03
Q13	/03
Q14	/03
Q15	/03
Q16	/05
Q17	/03
Q18	/08
TOTAL	/50

2025 Statistic	TOT/50	TOT%
Average:	*39.9*	*80%*
Median:	*42*	*84%*
Min:	*26*	*52%*
Max:	*47*	*94%*

Software Engineering

2025 Year 12 (HSC) Trial Exam SOLUTIONS

Total Marks: 50; Weighting: 25%

Duration: 1 hour & 30 minutes + 5 minutes reading

General Instructions:

- Write your student number at the top of this paper and all writing booklets
- Write your answers **CLEARLY and LEGIBLY** using a black (or blue) pen
- Draw diagrams using a pencil
- Attempt ALL questions
- Use of calculators is NOT permitted

There are TWO sections in this exam paper:

Section I: Objective-response questions
* 12 marks (allow about 18 mins)
* Indicate/write your answers on THIS paper

Section II: Short-answer questions
* 38 marks (allow about 72 mins)
* Use SEPARATE writing booklet(s)

Section I – Objective-Response Questions (12 marks)

Indicate/write your answer on THIS paper. (Different objective-response question types require different types of indication, e.g. circling for multiple-choice questions.)

Answer all questions.

Question	Q1/1	Q2/1	Q3/1	Q4/1	Q5/2	Q6/2	Q7/2	Q8/2
Average:	*1.0*	*0.4*	*0.9*	*0.8*	*1.6*	*1.9*	*1.4*	*1.5*
Average %:	*100%*	*36%*	*91%*	*82%*	*82%*	*95%*	*68%*	*73%*
Median:	*1*	*0*	*1*	*1*	*2*	*2*	*1*	*2*
Min:	*1*	*0*	*0*	*0*	*1*	*1*	*1*	*0*
Max:	*1*	*1*	*1*	*1*	*2*	*2*	*2*	*2*

Question 1 (1 mark)

A part of an application with a GUI requires from its user to answer several questions. For one type of question there are several correct responses, and the user must select all the correct ones from a set of five options. Which screen element would be best suited for this type of question?

[circle only 1 best answer]

A Radio buttons

B Command buttons

C Checkboxes

D List box

=> *Correct answer:* C (Checkboxes)

Marking criteria	***Marks***
Completely correct answer	**1**

***Marker's notes**:*

1) All students (100%) answered this question correctly.

Question 2 (1 mark)
Order the 4 steps in the model development part of MLOps:

[write a number 1 to 4 in front of each step; no repeats; 1 is executed first]

___ Model training

___ Data wrangling (preparation)

___ Model testing and validation

___ Feature engineering

=> *Correct answer:* 3, 1, 4, 2. This is because the order of the 4 steps given in the Course Specifications document is: 1) Data wrangling; 2) Feature engineering; 3) Model training; 4) Model testing and validation.

Marking criteria	***Marks***
Completely correct order of ALL given steps	**1**

Marker's notes*:*

1) *Feature engineering can be performed before Data wrangling (and some sources even list it as a part of Data wrangling), so this was also accepted on this internal Trial Exam (although it will probably not be accepted on the external HSC Exam).*
2) *This question was answered correctly only by a small number of students (36%). This indicates the need for students to learn the MLOps process as listed in the NESA Course Specifications for Software Engineering and, particularly, when Feature engineering happens. However, this poor result also indicates the need for students to practice more such ordering questions.*
3) *Many students put Feature engineering too late (while it MUST be done before Model Training). Since this is a 1-mark objective response question, any mistake results in 0 marks for this question.*

Question 3 (1 mark)
Which of the following best describes metadata?

[circle only 1 best answer]

A The primary content of a document or file

B Descriptive information about data that helps to organise, find, and understand it

C A summary of the given data that highlights its main statistics

D Encrypted data that is primarily used in social networking software like Facebook

=> *Correct answer:* B (Descriptive information about data that helps to organise, find and understand it)

Marking criteria	***Marks***
Completely correct answer	**1**

Marker's notes:

1) *Almost all students (91%) answered this question correctly. 1 student chose the distractor option D.*

Question 4 (1 mark)
How can SQL code be improved to prevent SQL injection attacks?

[circle only 1 best answer]

A By using user input directly in the SQL query with no sanitisation

B By using dynamic SQL queries that concatenate user input directly into the query

C By using only stored procedures with no parameterisation

D By implementing prepared statements with parameterised queries

=> *Correct answer:* D (By implementing prepared statements with parameterised queries)

Marking criteria	***Marks***
Completely correct answer	**1**

Marker's notes*:*

1) *Most students (82%) answered this question correctly. 1 student chose the distractor option C, while another 1 student chose the distractor option B.*

Question 5 (2 marks)

Which features of OOP can be seen in the following Python code?

```
class Person:
  def __init__(self, fname, lname):
    self.__firstname = fname
    self.__lastname = lname
  def printname(self):
    print(self.__firstname,self.__lastname.upper(), end="")
  def greet(self):
    print("G'day ", end="")
    self.printname()
    print()

class Student(Person):
  def __init__(self, fname, lname, sid, gradyear):
    super().__init__(fname, lname)
    self.__studentID = sid
    self.graduationyear = gradyear
  def greet(self):
    self.printname()
    print(", welcome to the Class of", self.graduationyear)
```

[tick ALL that apply]

___ Encapsulation

___ Inheritance

___ Instantiation

___ Polymorphism

=> *Correct answer:* Encapsulation, Inheritance and Polymorphism, but NOT Instantiation

Marking criteria	***Marks***
Completely correct answer with ALL relevant boxes ticked and NO other boxes ticked	**2**
Partially correct answer with at least 2 (out of 3) relevant boxes ticked, possibly with another box also ticked incorrectly	**1**

***Marker's notes**:*

1) *You had this question in your Year 11 Software Engineering Final (Yearly) Exam, but in a different format. This is an example of how a past question can be re-written in one of the new objective response question formats.*
2) *Most students (63%) answered this question correctly and got full 2 marks, while the others got the partial marks (1 out of 2). Errors included: not selecting Encapsulation (while the 2 classes encapsulate related data and subprograms), selecting Instantiation (while it is not in the given code), not selecting Polymorphism (while the method greet() is polymorphic).*
3) *Remember that in OOP a class encapsulates data (attributes) and related subprograms (methods), so if a class is specified in code (as above) then encapsulation is used. In the above code, there are also examples of the other meaning of OOP 'encapsulation' as 'information hiding' because __ at the start of a method name or an attribute name in Python denotes private access.*
4) *Some students who got full marks on the previous version of this question in the Year 11 Software Engineering Final (Yearly) Exam did not get full marks on this new version in*

the Year 12 Software Engineering Trial Exam. This might be because of the more difficult format or because they forgot OOP features and did not revise them.

5) *"Tick/Choose/Select ALL that apply" questions are difficult because for full marks it is necessary to tick ALL relevant boxes and NO other box. If you make any mistake, you cannot get full marks. Thus, you must be very careful when answering such questions. You will not know the boundary for partial marks when you answer a question of this type, but it is safe to assume that partial marks (e.g. 1 out of 2) are awarded only if your mistake is relatively minor.*
6) *In online exams, "Tick/Choose/Select ALL that apply" questions use rectangular boxes (check boxes), contrary to multiple-choice questions that use circles (radio buttons).*

Question 6 (2 marks)
Complete the text below by writing 1 word into each provided space. Each word you write must be 1 of the following: *Configuration, Data, Exception, Identity, Memory, Session*.

[fill in each blank with 1 of the given words]

_____ management catches unexpected problems in a program, helping to achieve a smooth recovery.

_____ management helps keep track of user interactions with a website/application.

_____ management helps ensure that computer's storage space is used wisely.

=> *Correct answer:* Exception, Session, Memory.

Marking criteria	***Marks***
Completely correct answer with ALL words chosen correctly	**2**
Partially correct answer with at least 2 words chosen correctly	**1**

Marker's notes:

1) *Almost all students (91%) answered this question correctly. 1 student wrote Configuration instead of Exception and got partial marks (1 out of 2).*

Question 7 (2 marks)

Which of the following machine learning models are NOT suitable for a classification problem?

[tick ALL that apply]

- ☐ Decision tree
- ☐ Neural network
- ☐ Linear regression
- ☐ Logistic regression
- ☐ K-nearest neighbours (KNN)
- ☐ Polynomial regression

=> *Correct answer:* Linear regression and Polynomial regression, but NOT any of the following: Decision tree, Neural network, Logistic regression, K-nearest neighbours (KNN)

Marking criteria	***Marks***
Completely correct answer with ALL relevant boxes ticked and NO other boxes ticked	**2**
Partially correct answer with the 2 relevant boxes ticked, but with some other boxes also ticked incorrectly	**1**

***Marker's notes**:*

1) *This question was answered correctly by a relatively small number of students (36%) who got full 2 marks, while the other students got partial marks (1 out of 2).*
2) *Many students selected Logistic regression – while it is called 'regression', it is actually used only for classification. Neural networks and KNN can be used for classification or for regression. Since this is a 2-mark objective response question, any mistake results in 1 mark for this question, while many mistakes result in 0 marks.*

Question 8 (2 marks)
For each of the given descriptions, indicate the most suitable network protocol:

[draw 8 lines connecting each protocol listed on the left with its 1 best description on the right]

SFTP •	• Sending email messages from a client to a server or between servers
POP 3 •	• Addressing and routing data packets between devices on a network
IP •	• Retrieving and managing emails directly on a mail server, allowing access from multiple devices
HTTPS •	• Ensuring reliable, ordered and error-checked delivery of data packets between devices
SMTP •	• Securely exchanging encrypted website data between a web browser and a web server
IMAP •	• Encrypting data transmitted over a network to ensure secure communication and protect against eavesdropping and tampering
TCP •	• Downloading emails from a mail server to a local client, typically deleting them from the server
TLS •	• Transferring files over a secure shell connection

=> ***Correct answer:***
SFTP --- Transferring files over a secure shell connection
POP 3 --- Downloading emails from a mail server to a local client, typically deleting them from the server
IP --- Addressing and routing data packets between devices on a network
HTTPS --- Securely exchanging encrypted website data between a web browser and a web server
SMTP --- Sending email messages from a client to a server or between servers
IMAP --- Retrieving and managing emails directly on a mail server, allowing access from multiple devices
TCP --- Ensuring reliable, ordered and error-checked delivery of data packets between devices
TLS --- Encrypting data transmitted over a network to ensure secure communication and protect against eavesdropping and tampering

Marking criteria	***Marks***
Completely correct answer with ALL relevant matches indicated	**2**
Partially correct answer with at least 5 relevant matches indicated correctly	**1**

Marker's notes*:*

1) *While HTTPS indirectly (through its use of TLS) has the property of "Encrypting data transmitted over a network to ensure secure communication and protect against eavesdropping and tampering" and TLS indirectly (when applied in HTTPS) has the property of "Securely exchanging encrypted website data between a web browser and a web server", the answer given above (written with the "---" symbol) is more precise and, thus, better. At an external HSC Exam, objective response questions will be marked automatically. It is unknown whether its HSC Exam writers will provide alternative answers to the auto-marking system. Therefore, always check that you provide the best possible answer to an objective response question. Otherwise, you will lose marks for an answer for which you can argue as being correct, but that the HSC Exam writers did not predict as correct.*

2) *Majority of students (55%) answered this question correctly and got full marks; many (36%) got partial marks (1 out of 2), but 1 student got 0 marks due to too many mistakes.*

End of Section I

Section II – Short-Answer Questions (38 marks)

Write your answers clearly & legibly in the SEPARATE writing booklet(s) provided.
Answer all questions.

Question	Q9/4	Q10/3	Q11/3	Q12/3	Q13/3	Q14/3	Q15/3	Q16/5	Q17/3	Q18/8
Average:	*2.8*	*2.0*	*2.6*	*2.4*	*2.7*	*2.7*	*2.7*	*4.0*	*1.9*	*6.5*
Average %:	*70%*	*67%*	*88%*	*79%*	*91%*	*91%*	*91%*	*80%*	*64%*	*82%*
Median:	*3*	*2*	*3*	*2*	*3*	*3*	*3*	*5*	*2*	*7*
Min:	*2*	*1*	*2*	*1*	*2*	*2*	*1*	*0*	*1*	*3*
Max:	*4*	*3*	*3*	*3*	*3*	*3*	*3*	*5*	*3*	*8*

Questions 9 and 10 both use the following algorithm:

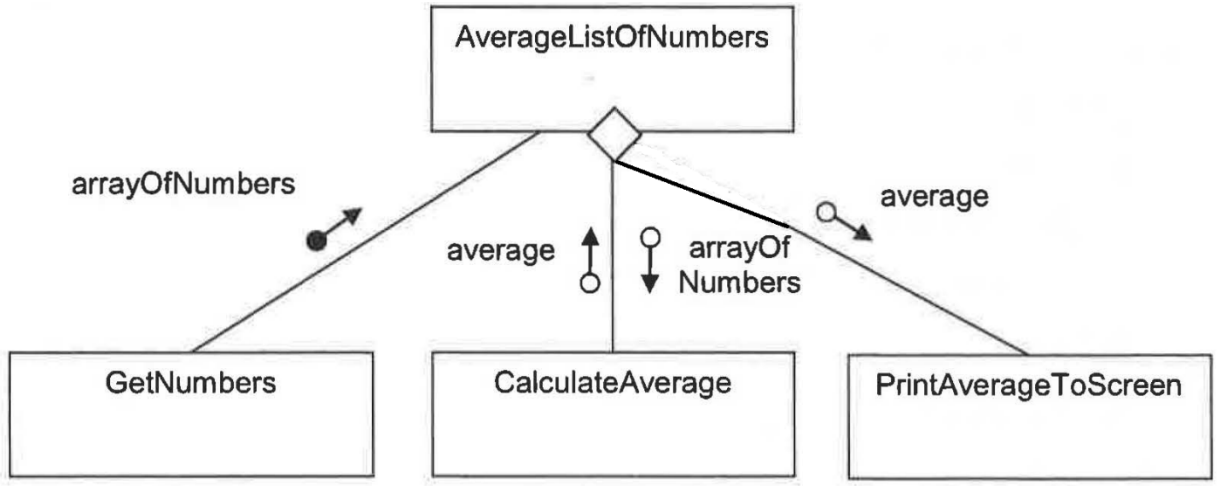
```
BEGIN AverageListOfNumbers
      average = -1
      arrayOfNumbers = GetNumbers()
      IF number of items in arrayOfNumbers != 0 THEN
            average = CalculateAverage(arrayOfNumbers)
            PrintAverageToScreen(average)
      ELSE
            PRINT "No numbers entered"
      END IF
END AverageListOfNumbers
```

Question 9 (4 marks)
Draw a structure chart to represent the `AverageListOfNumbers` algorithm given above.

[draw into the writing booklet the required diagram]

Marking criteria	*Marks*
Correct structure chart representing this algorithm	**4**
Substantially correct structure chart representing this algorithm, with some minor errors or omissions	**3**
An attempt at constructing a structure chart to represent this algorithm	**2**
An attempt at constructing a structure chart	**1**

Sample answer:



In this diagram, the "diamond" (rotated square) with 2 lines going from the same vertex denotes OPTIONAL execution of BOTH `CalculateAverage` *and* `PrintAverageToScreen` *subprograms (either BOTH of them are executed or NONE of them is executed). This is NOT a choice between executing either 1 of these 2 subprorams (a choice would be indicated by having the 2 lines coming from 2 different vertices of the "diamond"). The built-in commands like PRINT/DISPLAY and GET/INPUT are NOT shown on a structure chart.*

***Marker's notes**:*

1) *This question was NOT done well, which indicates that the students should revise structure charts! Only 18% of students got the full 4 marks, many (45%) got 3 out of 4, while the others got only 2 out of 4.*
2) *Many students showed on the structure chart processing that is NOT calls to subprograms. Such information is NOT to be shown on structure charts. Notably, the built-in commands like PRINT/DISPLAY and GET/INPUT are NOT shown on a structure chart. Similarly, processing such as "Initialise averageValue" is NOT to be shown on a structure chart. ONLY calls to subprograms (excluding PRINT/DISPLAY and GET/INPUT that are in some programming languages built-in commands rather than standard subprograms) are shown on a structure chart.*
3) *Several students drew PrintAvererageToScreen below CalculateAverage. However, this denotes that CalculateAverage CALLS PrintAverageToScreen, which is NOT the case in the given algorithm. Thus, CalculateAverage and PrintAverageToScreen MUST be shown on the same level (with CalculateAverage on the left), which indicates that they are called sequentially by a higher-level module (in this case: AverageListOfNumbers).*
4) *Many students did not identify arrayOfNumbers as a flag. While it is not a Boolean value (the usual data type for flags), arrayOfNumbers is a flag in this algorihm because its size determines which branch of the IF-ELSE-ENDIF statement is executed.*

Question 10 (3 marks)

For the `AverageListOfNumbers` algorithm given above, write pseudocode for the function `CalculateAverage(arrayOfNumbers)`. Assume that `arrayOfNumbers` contains only numbers. However, check whether `arrayOfNumbers` is empty, so that your function `CalculateAverage(arrayOfNumbers)` can be also reused in other contexts.

[write into the writing booklet the required pseudocode]

Marking criteria	***Marks***
Substantially correct algorithm	**3**
Partially correct algorithm, with some errors or omissions	**2**
An attempt at writing an algorithm to meet the requirements	**1**

Sample answer:

```
BEGIN CalculateAverage(arrayOfNumbers)
      count= LENGTH(arrayOfNumbers)
      IF count > 0 THEN
            total = 0
            FOR i = 1 TO count STEP 1  REM Assumes indexing from 1
                  total = total + arrayOfNumbers[i]
            NEXT i
            RETURN total / count
      ELSE
            RETURN Null REM No value; impossible to calculate average
      END IF
END CalculateAverage
```

***Marker's notes**:*

1) *This question was NOT answered well, indicating the need to revise pseudocode. Only 1 student (9%) got the full 3 marks. Most students (82%) got 2 out of 3 marks, while 1 student (9%) got only 1 out of 3 marks.*
2) *Several students used a FOR ... IN ... loop that does NOT exist in SOE pseudocode. The Course Specifications document lists clearly which constructs you are allowed to use in SOE pseudocode – learn them and use ONLY them!*

Question 11 (3 marks)
Explain how the capabilities of end users influence the secure design features of software.

[write into the writing booklet at least 5 information-rich sentences]

Marking criteria	*Marks*
Explains causal relationships between capabilities of end users and the secure design features of software	**3**
Discuses capabilities of end users relevant for secure design features of software	**2**
Identifies a capability of end users relevant for secure design features of software	**1**

Sample answer:

Capabilities of end users, such as their technical proficiency, security awareness and accessibility requirements, significantly impact the design features of software. They influence the user interface, accessibility options, customisation preferences and security settings. Developers must consider the capabilities of end users to create secure, usable, and effective software. (Additionally, they should provide appropriate documentation, training, and support to users with different capabilities.)

1) Technical Proficiency: End users have diverse levels of technical expertise, from beginners to advanced users. Software needs to be designed with intuitive interfaces and security settings (e.g. built-in prompts for secure actions) for less tech-savvy users, while providing advanced features for those who need them and can use them without introducing vulnerabilities.
2) Security Awareness: Users' understanding of security threats and security practices differs greatly. Software needs to account for the varying levels of security awareness among its users and mitigate risks associated with less security-conscious users. Commonly, this includes providing secure defaults (e.g. requiring multi-factor authentication by default), restricting unnecessary user permissions and displaying clear security prompts. Software designers often need to balance security enhancements with user experience, ensuring that security measures are as seamless as possible to encourage compliance and reduce the likelihood of users bypassing security for convenience.
3) Accessibility requirements: End users may have different accessibility requirements, influencing how security features are implemented. For instance, software design must ensure that security measures (e.g. multi-factor authentication) are accessible to users with disabilities, ensuring that security does not come at the cost of usability.

***Marker's notes**:*

1) This question was answered relatively well, with the majority (64%) of students getting full 3 marks and the others getting partial 2 out 3 marks.
2) Many students wrote general statements that are true, but do not answer the given question.
3) In particular, several students did not consider influence of different user capabilities (e.g. the 3 listed in the sample answer).

Question 12 (3 marks)

Describe what a template engine in web development is and how it works.

[write into the writing booklet at least 5 information-rich sentences]

Marking criteria	***Marks***
Describes both what a template engine is AND how it works	**3**
Describes either what a template engine is OR how it works	**2**
States a feature of a template engine	**1**

Sample answer:

A template engine (a.k.a. templating engine) is used to rapidly and easily build web applications and dynamic HTML pages. It leverages HTML/JavaScript templates (some are pre-made but configurable, others can be written by designers) that contain placeholders, including variables, loops iterating over collections and conditionals. Different template engines (e.g. EJS, Pug, Handlebars, etc.) use different notations (syntaxes) to describe their templates and placeholders. When the web application is running, the template engine on the web server replaces the variables/placeholders in the used template file with actual values, then sends this dynamically customised HTML web page content to the client (which never sees the used templates). Templates also enable fast rendering of server-side data that needs to be passed to the application. By using a template engine, developers can separate HTML presentation structure from the application logic, making the code cleaner and easier to maintain. Designers can work on templates while developers focus on the backend logic.

***Marker's notes**:*

1) *The number of students got the full 3 marks or 2 out of 3 marks was equal (45%). However, 1 student got only 1 mark out of 3.*
2) *Several students did not explain how the template engine works, particularly how it replaces placeholders with concrete data values during runtime (i.e. dynamically).*
3) *Several students only described static (development-time) template customisation to create web sites. This can be done by using templates provided with template engines, but it is NOT the main feature of template engines. It is a characteristic of some other web development systems that use static web page templates allowing non-developers to create web sites, but these are NOT template engines.*

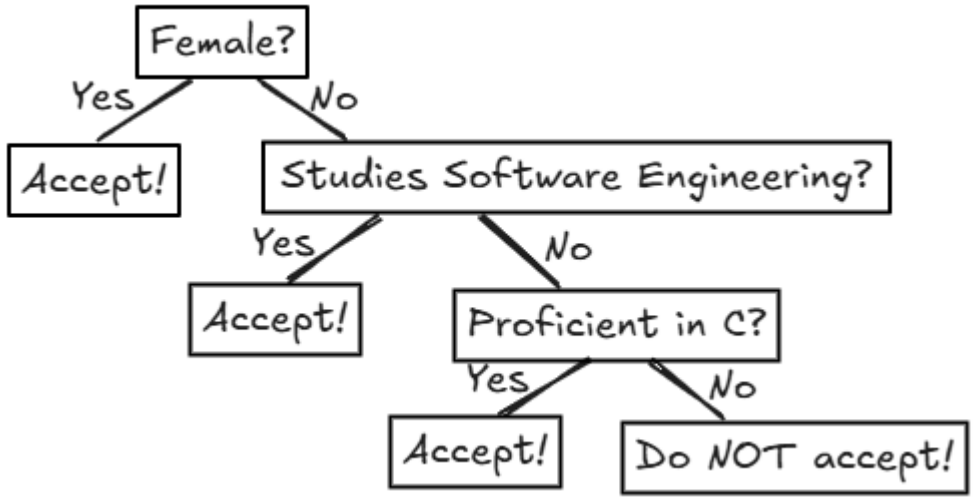
Question 13 (3 marks)

Draw a decision tree (resulting from machine learning) for decisions whether a person should be accepted into a software engineering student club meeting. A female is always accepted. Other persons are accepted only if they study Software Engineering or are proficient in C.

[draw into the writing booklet the required diagram]

Marking criteria	*Marks*
Substantially correct decision tree	**3**
Partially correct decision tree, with some errors or omissions	**2**
An attempt at drawing a decision tree to meet the requirements	**1**

Sample answer:



***Marker's notes**:*

1) *This question was answered well. Most students (73%) answered this question well and got the full 3 marks, while the others got 2 out of 3.*
2) *Remember that in a decision tree "each node (other than a leaf) represents a decision [point], each branch link represents a possible outcome of the decision in the originating node, and each leaf node represents a class label [or a final outcome/action]". Several students drew a tree-like structure that does not contain decisions (questions, conditions) in nodes, but placed them in branch labels – this is not an appropriate representation of a decision tree.*
3) *The Course Specifications document does not prescribe particular symbols to use in decision trees, but its example on page 10 uses rectangles for all nodes. Sources on the internet use different symbols (e.g. rounded examples or ovals), particularly for non-leaf nodes. While you can use different symbols, it seems safer to use the notation provided in the Course Specifications document – rectangles for all nodes.*
4) *For practical reasons (e.g. computational complexity), most decision trees resulting from machine learning are binary, i.e. with only 2 branches coming from a decision node. The 2nd example on page 10 of the Course Specifications document is a binary tree where all branches labelled either Yes or No. If you are asked to draw a decision tree on the external HSC Exam, try drawing (if possible) a binary decision tree where all branches are labelled either Yes or No.*
5) *In the online exams (most importantly: the external HSC Exam), this type of question will require use of the online platform's drawing tool and its library of built-in symbols. Practise using them!*

Question 14 (3 marks)

Compare and contrast cross-site scripting (XSS) and cross-site request forgery (CSRF).

[write into the writing booklet at least 5 information-rich sentences]

Marking criteria	***Marks***
Discusses at least 2 similarities AND at least 2 differences between XSS and CSRF	**3**
Discusses either ONLY similarities OR ONLY differences between XSS and CSRF OR Outlines a similarity AND a difference between XSS and CSRF	**2**
Provides some correct information about XSS or CSRF	**1**

Sample answer:

Similarities between XSS and CSRF include:

1) Both XSS and CSRF are vulnerabilities that specifically target web applications.
2) Both attacks rely on exploiting the trust that users and web applications have, particularly the inherent trust a website has in its users and their browsers.
3) Both vulnerabilities can lead to client-side consequences, affecting what a user can see or do on a web application.
4) Both XSS and CSRF can lead to serious consequences, including access to sensitive information, data breaches, account hijacking, and website defacement.

Other similarities: the need for mitigation, the need for user awareness.

Differences between XSS and CSRF include:

1) Different focus: While XSS primarily targets the user's browser and client-side security, CSRF primarily targets the server and authentication mechanisms.
2) Different attack mechanisms (vectors): XSS involves injecting malicious scripts (typically JavaScript) into a website, which then executes in the user's browser when they visit the compromised page. Contrary, CSRF involves an attacker crafting a malicious link or form that, when clicked or submitted by the unsuspecting user, sends a request to the target website as if it were from the user themselves.
3) Different data handing: In XSS the attacker can potentially read the response of the malicious script's execution and extract data to their own server. In CSRF, the attacker can only send requests, but they cannot directly receive the responses from the target website.
4) Different session requirement: XSS can be successful regardless of the user's session status (authenticated or not), but CSRF requires the user to have an active session with the targeted website.

Other differences: different prevention methods.

***Marker's notes**:*

1) *Most students (73%) answered this question well and got the full 3 marks. The other students who got 2 out of 3 did not provide sufficient details (at least 2 similarities and at least 2 differences), plus 1 student was not precise enough in his explanations.*

Question 15 (3 marks)

An online business is planning to use a database to keep track of its products. The contents of the Products table during testing are shown.

Products

ProductID	ProductName	ProductPrice
P001	The Plant (DVD)	28.00
P002	Discovery 1 and 2 (DVD)	26.98
P003	Travel 1 and 2 (Blu-Ray)	22.00
P004	The Best Movie (DVD)	19.98
P005	Celebration (DVD)	12.00

After executing a SQL query, the following results were obtained:

ProductName	ProductPrice
Celebration (DVD)	12.00
The Best Movie (DVD)	19.98
Travel 1 and 2 (Blu-Ray)	22.00

Write the SQL SELECT query that produced the above results.

[write into the writing booklet the required SQL query]

Marking criteria	*Marks*
Substantially correct SQL SELECT statement	**3**
Partially correct SQL select statement, with some errors or omissions	**2**
An attempt at writing a SQL SELECT statement to meet the requirements	**1**

Sample answer:

```
SELECT ProductName, ProductPrice
FROM Products
WHERE ProductPrice <= 22.00
ORDER BY ProductPrice ASC;
```

An alternative sample answer:

```
SELECT ProductName, ProductPrice
FROM Products
WHERE ProductPrice <= 22.00
ORDER BY ProductName ASC;
```

***Marker's notes**:*

1) *This question was answered well. Most students (82%) got the full 3 marks, 1 student got 2 out 3, but 1 student got only 1 out 3.*
2) *Simple SQL SELECT statement questions are more likely to be given as objective-response questions with missing parts selected using drop-down boxes. Compared to a short answer question (like the one given in this exam), an objective-response question is somewhat easier so it might be worth fewer marks (e.g. 2 instead of 3).*
3) *This is a relatively simple SQL SELECT statement. However, the Software Engineering Syllabus also specifies that you need to know table joins and GROUP BY clauses (the latter is usually used with an aggregate function). There are different types of table joins and it is*

unclear from the Syllabus whether you need to know all of them. In any case, inner joins are the most important type – they are specified using the keyword JOIN (or INNER JOIN or, in some SQL versions, KEY JOIN). Table join questions can be difficult, so practice them additionally.

4) *In the given question, choosing ORDER BY ProductName or ProductPrice would produce the same result table. This was a pure coincidence, but it seems that it confused some students. One student made the mistake of not providing a column name in the ORDER BY clause. Note that an ORDER BY clause MUST list at least 1 column name (if several column names are provided, the 2nd column is used for rows that have the same values in the 1st column, etc.). Another student did not write the ORDER BY clause at all – this results in rows in a random order (while the given question shows a sorted order).*

Question 16 (5 marks)

An ISBN is the International Standard Book Number, a globally unique number to identify books. This is usually printed above the barcode on the back cover of most books.

The last (i.e. the 13th) digit in a 13-digit ISBN (International Standard Book Number) is known as the "check digit". The check digit is calculated in the following way:

- Each of the other 12 digits of the ISBN from left to right is multiplied alternately by 1 or 3. This means that the first digit is multiplied by 1, the second by 3, the third by 1, the fourth by 3, etc.
- The results are added together and then divided by 10.
- The remainder of this division is subtracted from 10 to give the check digit.
- If the check digit is 10 then it is replaced with 0 – this ensures that the check digit is always a single digit.

For example, in an ISBN of 9780980874921 the last digit (in this case 1) is the check digit. However, we should re-calculate that check digit to ensure that the ISBN has not been entered incorrectly. (9x1) + (7x3) + (8x1) + (0x3) + (9x1) + (8x3) + (0xl) + (8x3) + (7xl) + (4x3) + (9xl) + (2x3) = 129. Then, 129 / 10 = 12 remainder 9. Therefore, the check digit is 10 - 9 = 1. In this case the ISBN is valid as our calculated value of 1 matches the provided check digit.

In Python, write the function `checkDigitOK(isbn_str)` that takes a 13-digit ISBN as the string `isbn_str` and returns `True` or False indicating whether the ISBN is valid in terms of its check digit. Assume that the provided string `isbn_str` always contains 13 digits – do NOT check whether there are fewer/more digits and do NOT check whether some of the characters are not digits. Do NOT write the main program nor any other subprogram calling `checkDigitOK(isbn_str)`.

Hints:

- The 13-digit ISBN is provided as a string, so conversion of individual characters into integer digits is needed for the calculations.
- You can use the Python's `%` operator to calculate the remainder of a division.

[write into the writing booklet the required Python code]

Marking criteria	***Marks***
Correct Python program that includes ALL of the following 5 features: • Loops through each ISBN digit (except the last) and works out the sum of digits • Correctly distinguishes between digits to multiply by 3 and add vs those to just add • Correctly calculates the check digit • Performs a check to see whether the check digit is 10 and replaces it with 0 • Compares the calculated check digit to the last digit of the entered ISBN and returns True if they match (correct check digit) or False if they do not (incorrect check digit)	**5**
Substantially correct Python program that includes at least 4 of the above features	**4**
A sound attempt at the Python program that includes at least 3 of the above features	**3**
An attempt at writing the Python program that includes at least 2 of the above features	**2**
An attempt at writing the Python program that includes at least 1 of the above features	**1**

Sample answer:

```
#Checks whether the check digit of an ISBN is valid
def checkDigitOK(isbn_str): #assumes a string with 13 digits provided
  sumOfDigits = 0 #accumulator for the sum of digits
  for i in range(12): #loops thru the first 12 digits but not the last/13th
    #1st digit has Python index 0 (even); 2nd digit has index 1 (odd), etc.
    if i % 2 == 0: #an even Python index, the digit is multiplied by 1
      sumOfDigits = sumOfDigits + int(isbn_str[i]) * 1
    else: #an odd Python index, the digit is multiplied by 3
      sumOfDigits = sumOfDigits + int(isbn_str[i]) * 3
  sumModulo = sumOfDigits % 10 #calculates the remainder of division by 10
  checkDigit = 10 - sumModulo #calculates the check digit
  if checkDigit == 10: #the special case; checkDigit must have only 1 digit
    checkDigit = 0
  #Checking whether the provided ISBN contains the correct check digit
  if int(isbn_str[-1]) == checkDigit: #last element is at Python index -1
    return True
  else:
    return False
```

An alternative sample answer:

```
#Checks whether the check digit of an ISBN is valid
def checkDigitOK(isbn_str): #assumes a string with 13 digits provided
  sumOfDigits = 0 #accumulator for the sum of digits
  #the first 12 digits of ISBN contain 6 groups of 2 digits:
  #   the 1st of these 2 digits is multiplied by 1
  #   the 2nd of these 2 digits is multiplied by 3
  for i in range(6):
    sumOfDigits = sumOfDigits + int(isbn_str[2*i]) * 1
    sumOfDigits = sumOfDigits + int(isbn_str[2*i+1]) * 3
  sumModulo = sumOfDigits % 10 #calculates the remainder of division by 10
  checkDigit = 10 - sumModulo #calculates the check digit
  if checkDigit == 10: #the special case; checkDigit must have only 1 digit
    checkDigit = 0
  #Checking whether the provided ISBN contains the correct check digit
  if int(isbn_str[12]) == checkDigit: #last/13th digit is at Python index 12
    return True
  else:
    return False
```

Marker's notes:

1) *Most students did this question relatively well. The majority (55%) got the full 5 marks, some (18%) got 4 out of 5 and the same number got 3 out of 5. Unfortunately, 1 student completely missed this question (left space in the booklet, but never came back to it – probably due to the lack of time) and got 0 out of 5.*
2) *In the external HSC Exam, you will NOT be given hints like the ones you were given in this question. You are supposed to be proficient Python programmers by now, knowing about various aspects of the language, including type conversions and the % operator. Nevertheless, you were provided a couple of hints in this internal Trial Exam to lower your stress when writing answers to this somewhat complicated (5 mark) question.*
3) *A surprisingly large number of students used = instead of == for comparisons in Python, e.g. "if sum = 0:" instead of "if sum == 0:" . Less frequent Python syntax mistakes included: writing TRUE and FALSE instead of True and False, forgetting ":", forgetting "def", writing =+ instead of +=, writing x instead of * for multiplication WS… If you are asked to write a Python program, you MUST use Python syntax specifics. While in the online HSC Exam you will have feedback about syntax errors, you will then lose time to debug and fix these syntax errors. Learn Python syntax well and apply it the first time you write your program!*

4) *A few students missed checking the special case "if checkDigit == 10:". When writing an algorithm in pseudocode or writing a program in a programming language based on a given textual description, always re-check that your algorithm/code covers all required steps described in the textual description.*
5) *Writing each of the first 12 digits individually to sum them (e.g. sumOfDigits = int(isbn_str[0])*1+int(isbn_str[1])*3+int(isbn_str[2])*1+int(isbn_str[3])*3+...) is too specific (e.g. might require bigger modifications for any future ISBN with more than 13 digits). Instead, a pattern should be noticed and a loop should be used to iterate through the first 12 digits. Nevertheless, writing each of the first 12 digits individually to sum them was accepted in this internal Trial Exam (although it should be avoided at the external HSC Exam).*

Questions 17 and 18 both use the following scenario:

Your team is developing the secure PWA 'nAIs Skin Check PWA'. Using this app, a user first uploads images of their skin moles, personal information (e.g. name, phone number, email, …), demographic information (e.g. age, gender, occupation, …) and medical history. Then, the app uses machine learning / artificial intelligence to determine and show on the screen the probability of melanoma (dangerous skin cancer) in the given skin mole images.

Question 17 (3 marks)

Discuss how the 3 (THREE) main principles of the 'privacy by design' approach should be applied in the development of the 'nAIs Skin Check PWA' app (the scenario described above).

[write into the writing booklet at least 5 information-rich sentences]

Marking criteria	***Marks***
Discusses applying on the given scenario of 3 of the main principles of 'privacy by design'	**3**
Discusses applying on the given scenario of 1 or 2 of the main principles of 'privacy by design' OR Outlines 2 or 3 of the main principles of 'privacy by design'	**2**
Shows some understanding of 'privacy by design' OR Outlines 1 of the main principles of 'privacy by design'	**1**

Sample answer:

The 'privacy by design' is built around the 7 main principles:
1. Proactive not reactive, preventative not remedial
2. Privacy as a default setting
3. Privacy embedded into design
4. Full functionality: positive-sum not zero-sum
5. End-to-end security - full lifecycle protection
6. Visibility and transparency - keep it open
7. Respect for user privacy - keep it user centric

From these 7, our Software Engineering syllabus emphasises only #1, #3 and #7 so their application on the given 'nAIs Skin Check PWA' app scenario is discussed here.

Proactive not reactive approach is about anticipating risks and preventing privacy-invasive events before they occur. In the 'nAIs Skin Check PWA' app scenario, the personal and medical information (possibly also some of the demographic information) is personally identifiable information (PII) that must be kept private and protected. It can be anticipated that this sensitive information could be compromised on the client side, transport over the Internet or the server side. Therefore, all information provided by a user should be protected (e.g. encrypted) in transport over the Internet and any client-side and server-side handling, including storage. This is protection in transit, in use and rest. When requirements for the 'nAIs Skin Check PWA' app are determined, it should be decided whether it is necessary to keep any user information (e.g. as user accounts) on the server or the client – as an individual user would use this app relatively rarely, it is better to require that 'nAIs Skin Check PWA' does NOT store any user information (neither on the server nor on the client) after the probability of melanoma in the given images is shown to the user. It can be also anticipated that user information could be compromised by being

provided (directly or indirectly) to a 3rd party. Thus, it must be required that none of this information should be made available to any 3rd party.

Embed privacy into design is about ensuring that privacy becomes one of the core functions of any system or service. In the 'nAIs Skin Check PWA' app scenario, this means the app's design must include the (previously-mentioned) decisions to protect all user's information in transport/storage/handling, not storing any user information (neither on the server nor on the client) after a user is shown the probability of melanoma in the given images, and not providing any user information to any 3rd party.

Respect for user privacy is about keeping the interest of individuals paramount for any system or service, e.g. by offering strong privacy defaults and user-friendly options, as well as ensuring appropriate notices are given. In the 'nAIs Skin Check PWA' app scenario, the default protection of all user information in transit/rest/use, the default server-side and client-side secure deletion/destruction of all user information, and the default of not providing of any user information to any 3rd party are strong with respect to user privacy. As an individual user would use this app relatively rarely, it seems better NOT to give users an option to modify these privacy defaults. The app should also give users appropriate notices about how their information is used and protected.

Marker's notes:

1) *This question was NOT answered well, indicating the need to additionally revise 'privacy by design'. Only 1 student got the full 3 marks, most (73%) got 2 out of 3, while 2 students got only 1 out of 3.*
2) *Discussing ANY 3 of the 7 main principles of 'privacy by design' is awarded marks on this internal exam/test because you have been provided videos and online articles that discuss all 7 of them. However, on any external HSC Exam you should focus on the 3 principles listed in NESA's Software Engineering syllabus: proactive not reactive approach, embed privacy into design and respect for user privacy.*
3) *Several students identified the "CIA triad" of confidentiality, integrity and availability as the 3 main principles of 'privacy by design'. Although they (mostly confidentiality) are somewhat related to 'privacy by design', these are not the principles that this question asked about. Remember also that security and privacy are related, but are NOT the same.*
4) *Almost all students did not explicitly name any of the 7 main principles of 'privacy by design', but they did discuss relevant characteristics of 'privacy by design' and applied them on the given scenario. Such answers could not get full marks, but they were given partial marks (2 or 1 depending on the depth of the discussion).*

Question 18 (8 marks)

Explain, illustrate on examples from the 'nAIs Skin Check PWA' scenario (described above) and evaluate strategies that your software development team should use to manage security of programming code, including: code review, SAST, DAST, vulnerability assessment, penetration testing.

[write into the writing booklet at least 12 information-rich sentences]

Marking criteria	***Marks***
Explains and evaluates all 5 listed strategies for managing security of programming code AND supports this with examples from the given scenario	**8**
Explains and evaluates some of the 5 listed strategies for managing security of programming code (while only discussing the others OR explaining them but without evaluating them OR explaining & evaluating them but with errors) AND supports this with examples from the given scenario	**7**
Discusses all 5 listed strategies for managing security of programming code AND supports this discussion with examples from the given scenario	**6**
Discusses some of 5 listed strategies for managing security of programming code (while only outlining the others or discussing them but with errors) AND supports this discussion with examples from the given scenario	**5**
Correctly outlines all 5 listed strategies for managing security of programming code	**4**
Correctly outlines some of the 5 listed strategies for managing security of programming code	**3**
Shows some understanding of strategies for managing security of programming code	**2**
Provides some relevant information	**1**

Sample answer:

Code review, Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), vulnerability assessments, and penetration testing are all valuable strategies for managing the security of programming code, but they serve different purposes and are most effective when used in combination.

Code review is a manual (or partially automated) process of systematically examining source code to identify security flaws, logic errors, other mistakes or quality issues, and adherence to coding standards. It is usually performed by 1 or several software engineers different from the writer of the reviewed code. In the 'nAIs Skin Check PWA' app scenario, this would mean that some team members systematically review code written by other team members and provide improvement feedback. Code review benefits include: encourages knowledge sharing, catches vulnerabilities and bugs early in the development cycle, reduces the cost of remediation, and enhances both code quality and security. Its limitations include: can be time-consuming, may not identify all runtime vulnerabilities, may become less effective when not done regularly.

Static Application Security Testing (SAST) is a white-box testing method that analyses source code (less often: bytecode or binary code) for security vulnerabilities without executing the program. In the 'nAIs Skin Check PWA' app scenario, this would mean that some team members are trained in SAST and in using particular SAST tools that they would use to perform SAST on the app's code without executing it, providing improvement feedback to the team. SAST benefits include: can identify vulnerabilities and issues (such as SQL injection, buffer overflows, insecure APIs, insecure coding practices) early in the development process. Its limitations include: can

produce false positives (leading to time spent investigating benign issues), may not detect runtime-specific vulnerabilities.

Dynamic Application Security Testing (DAST) is a black-box testing method that runs software and analyses its behaviour (e.g. when exposed to simulated attacks) to find vulnerabilities (such as authentication bypasses or session management flaws) that are only apparent during runtime, i.e. that an attacker could exploit in a live environment. In the 'nAIs Skin Check PWA' app scenario, this would mean that some team members are trained in DAST and in using particular DAST tools that they would use to perform DAST on the running app (with simulated attacks), providing improvement feedback to the team. DAST benefits include: identifies real-world runtime vulnerabilities (that may not be caught by SAST), can find issues related to configuration and runtime environments, provides insights into how the software behaves under attack, helps ensure that security defences work as expected in a live environment. Its limitations include: typically occurs later in the development cycle as it may require a fully deployed application (this delays feedback), can be limited by the scope of the tests performed, may not detect vulnerabilities in the code itself, can be more time-consuming than SAST.

Vulnerability assessment is a systematic process of identifying, quantifying, classifying and prioritizing security vulnerabilities in an application or system. It often includes scanning for known vulnerabilities and assessing the security posture of the application. In the 'nAIs Skin Check PWA' app scenario, this would mean that some team members are trained to use appropriate tools to systematically scan for, identify and prioritise security vulnerabilities in the app and assess the security posture of the app, then provide improvement feedback to the team. This process is often outsourced to or guided by external experts. Vulnerability assessment benefits include: provides a comprehensive overview of vulnerabilities and software's security posture, helps prioritise remediation efforts based on risk levels, can be used to assess the effectiveness of security controls, is essential for maintaining an ongoing security strategy. Its limitations include: may not identify all vulnerabilities (especially those that require dynamic analysis or specialised expertise), may overlook newly discovered vulnerabilities or custom code weaknesses that are not part of standard libraries, can become outdated quickly if not performed regularly.

Penetration testing is performing simulated real-world attacks on a system or application to identify vulnerabilities that can be exploited in the software or the underlying infrastructure and to assess the effectiveness of security controls. In the 'nAIs Skin Check PWA' app scenario, this would mean that external hired white-hat hackers (less often: some team members trained in penetration testing) perform simulated attacks on the running app, identifying vulnerabilities, and providing improvement feedback to the team. Penetration testing benefits include: provides a realistic real-world assessment of the security posture, identifies vulnerabilities that may not be detected by other methods, helps organizations understand their risk profile, often results in detailed reports with actionable remediation steps. Its limitations include: can be time-consuming, can be resource-intensive and expensive, requires specialised expertise by skilled professionals, it is a point-in-time assessment so ongoing testing is needed to maintain security.

Incorporating a combination of these strategies provides a robust approach to managing code security. Code reviews and SAST are essential for early detection, while DAST, vulnerability assessments, and penetration testing provide insights into a deployed application's security. By integrating these methods into the software development lifecycle, teams (such as the 'nAIs Skin Check PWA' app team) can significantly enhance their security posture and reduce the risk of vulnerabilities being exploited.

Marker's notes:

1) *This question was done relatively well (resulting in relatively high albeit not full marks), but every student has some aspects for improvement. While only 27% of students got the full 8 marks, the most common mark was 7 out of 8 – 45% of students got it.*
2) *Many students did not fully (or even at all) address the 'evaluate' requirement of this question. This was the most common reason for not getting full marks.*

3) *Unfortunately, 1 student missed the point of this question – although he provided application on the given scenario, his answer only sporadically mentions the 5 strategies instead of focusing on them. This resulted in only 3 out 8 marks.*
4) *It is important to explain that "static" in SAST means WITHOUT executing/running the code, while "dynamic" in DAST means WITH executing/running the code. Several students have not done this, lowering the quality of their discussion of SAST and DAST.*
5) *It is important to clarify that vulnerability assessment prioritises found vulnerabilities so that limited time and resources can be spent most efficiently. Several students have not done this.*

End of Paper