

NSW Independent Trial Exam
Software Engineering Year 12 Trial HSC Examination 2025

SUGGESTED ANSWERS AND MARKING CRITERIA

Section I – Objective Response (20 marks)

Multiple choice (8 marks)

1 D	2 A	3 A	4 B	5 C	6 C	7 B	8 B
---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------

Multiple answer (4 marks)

9 B, C, E, G	10 A, D, E, F, H	11 B, D, G, H	12 B, D, E, G
------------------------	----------------------------	-------------------------	-------------------------

Matching (8 marks)

13 1C 2A 3A 4B 5C 6B 7A	14 1D 2C 3E 4B 5A	15 1D 2C 3A 4D 5C 6B	16 1B 2C 3A 4D 5B 6A 7D 8C
---	---	---	---

Question 13 2 marks

Answer:

	Dynam ic Applic ation Securit y Testing (DAST)	Static Applica tion Securit y Testing (SAST)	Penet ration Testi ng
Simulates cyber attacks to find and exploit vulnerabilities.	☐	☐	☒
Requires either the source code or binary.	☐	☒	☐
Checks for security vulnerabilities without executing the code.	☐	☒	☐
Evaluates environment and runtime issues.	☒	☐	☐
Utilises scanning tools to understand how a target responds to intrusions.	☐	☐	☒
Checks for security vulnerabilities while the program is running.	☒	☐	☐
A purely white box testing strategy	☐	☒	☐

Question 14 2 marks**Answer:**

	Web Security	Machine- readable data	Privacy	Web Accessi bility Initiative (WAI)	Inter nati onal isati on
Creates standards and working groups to protect user data and privacy	☐	☐	✓	☐	☐
Develops standards and guidelines for secure web applications	✓	☐	☐	☐	☐
Defines open formats (like XML) for data to be processed by computers	☐	✓	☐	☐	☐
Supports web standards for multiple languages and cultural contexts	☐	☐	☐	☐	✓
Ensures web content is usable by people with disabilities	☐	☐	☐	✓	☐

Question 15 2 marks**Answer:**

	Semi-supervised learning	Supervised learning	Unsupervised learning	Reinforcement learning
Training a robot to navigate a maze	☐	☐	☐	☒
Labelling a small set of medical images and using them to classify a larger set	☒	☐	☐	☐
Predicting student grades based on past results	☐	☒	☐	☐
Teaching an AI agent to win a game through trial and error	☐	☐	☐	☒
Classifying email as spam or not spam	☐	☒	☐	☐
Grouping customers based on shopping habits	☐	☐	☒	☐

Question 16 2 marks**Answer:**

	Identifying and defining	Researching and planning	Producing and implementing	Testing and evaluating
Decide on a suitable software development approach.	☐	☒	☐	☐
Develop, construct and document algorithms.	☐	☐	☒	☐
Explore tools used to develop ideas and generate solutions.	☒	☐	☐	☐
Use a language-dependent code optimisation technique.	☐	☐	☐	☒
Explore social and ethical issues associated with project work.	☐	☒	☐	☐
Define and analyse the requirements of a problem.	☒	☐	☐	☐
Compare actual with expected output	☐	☐	☐	☒
Code the solution in a programming language	☐	☐	☒	☐

Section II – Short Answer (60 marks)

Secure Software Architecture

Question 17

Marking Criteria	Marks
Clear and concise explanation of both authentication and authorisation, effectively distinguishing between the two concepts. Includes relevant examples or details that enhance understanding.	2
Basic explanation of the concepts with some understanding demonstrated. May lack clarity in distinguishing between the two or omit examples.	1

Authentication and authorisation are two fundamental concepts in the field of security in software engineering, and while they are often used together, they serve distinct purposes.

Authentication is the process of verifying the identity of a user or system. It ensures that individuals or entities are who they claim to be before granting access to resources or systems. Common methods of authentication include the use of usernames and passwords, biometric verification (such as fingerprints or facial recognition), and multi-factor authentication (MFA), which combines multiple verification methods. For instance, when a user logs into their email account by entering their username and password, the system performs authentication to confirm that the user is legitimate and allowed to access that account.

On the other hand, authorisation is the process that determines what an authenticated user is allowed to do within a system. Once a user's identity has been verified through authentication, authorisation governs their access rights and permissions to various resources or functionalities. For example, in a company's internal database, a manager may have the authorisation to view and modify employee records, while a regular employee might only have the authorisation to view their own record. Authorisation ensures that users can only perform actions that they are permitted to do, thereby protecting sensitive data and maintaining system integrity.

Question 18

Marking Criteria	Marks
Explains race conditions clearly, describes how it affects the python function, and explains how the lock prevents it with accurate reference to inventory accuracy.	3
Explains race conditions and how the lock helps prevent it, though lacks detail or clarity.	2
Demonstrates some understanding of race conditions.	1

A race condition occurs when multiple threads or processes access and modify shared data at the same time, leading to unpredictable or incorrect outcomes. In this scenario, race conditions are prevented by using a locking mechanism (with lock:), which ensures that only one thread can access and modify the inventory at a time. This prevents two users from purchasing the same item simultaneously when only one is available. Without the lock, multiple users could trigger the condition at the same time, causing the inventory to drop below zero and resulting in inaccurate stock levels and unfulfillable orders.

Question 19

Marking Criteria	Marks
Clearly describes how confidentiality, accountability and integrity has been compromised.	4
Describes how confidentiality, accountability and integrity has been compromised.	3
Describes how TWO of confidentiality, accountability and integrity has been compromised.	2
Provides a response with some relevance to compromised confidentiality, accountability or integrity.	1

The school has experienced a fundamental violation of confidentiality, accountability and integrity:

- Confidentiality: A directory traversal attack has allowed students to access restricted files which compromises confidentiality. This violates the principle of ensuring data is only accessible to authorised users.
- Accountability: The ability to re-upload work without proper logging has compromised accountability. Without audit logs to track previous assignment submissions, it becomes difficult to trace changes over time which is essential for maintaining accountability in systems.
- Integrity: As students are able to upload their work after the due date without detection the integrity of the system is compromised. Integrity requires data to not only be correct but to remain correct over time which is not the case in this scenario.

Question 20

Marking Criteria	Marks
Provides a comprehensive discussion of the benefits of collaboration in software engineering, clearly highlighting three key advantages. Each advantage is well-explained with relevant details and examples. Clearly identifies a relevant collaboration tool and provides a detailed explanation of how it facilitates collaboration among developers, demonstrating a strong understanding of its features and benefits. Response is well-organised, logically structured, and easy to follow, enhancing the clarity of the discussion.	6
Discusses the benefits of collaboration with some detail, but may only partially explain three advantages or lacks depth in the explanations. Identifies a collaboration tool but provides a vague or incomplete explanation of how it facilitates collaboration; may lack specific details. Response is well-organised, logically structured, and easy to follow, enhancing the clarity of the discussion.	5
Discusses the benefits of collaboration with some detail, but may only partially explain three advantages or lacks depth in the explanations. Identifies a collaboration tool but provides a vague or incomplete explanation of how it facilitates collaboration; may lack specific details. Response is poorly organised or difficult to understand.	4
Mentions some benefits of collaboration, but lacks clarity and detail; may only identify one or two advantages without sufficient explanation. Identifies a collaboration tool but provides a vague or incomplete explanation of how it facilitates collaboration; may lack specific details. Response is poorly organised or difficult to understand.	3

Does not provide relevant information on the benefits of collaboration or presents significant inaccuracies. Identifies a collaboration tool but provides a vague or incomplete explanation of how it facilitates collaboration; may lack specific details. Response is poorly organised or difficult to understand.	2
Does not provide relevant information on the benefits of collaboration or presents significant inaccuracies. Does not provide a relevant example of a collaboration tool or provides inaccurate information. Response is poorly organised or difficult to understand.	1

Collaboration is essential in software engineering, as it significantly enhances the efficiency and quality of software development projects. Effective collaboration can yield several key advantages, including improved problem-solving, enhanced communication, and increased productivity.

Firstly, collaboration fosters improved problem-solving. When team members work together, they can share diverse perspectives and expertise, leading to innovative solutions to complex problems. For instance, a developer may encounter a challenging bug within the code. By collaborating with colleagues, they can brainstorm potential fixes and draw on each other's experiences, resulting in faster resolution and enhanced code quality.

Secondly, enhanced communication is a critical benefit of collaboration. Open channels of communication ensure that all team members are on the same page regarding project goals, requirements, and progress. This transparency helps to reduce misunderstandings and misalignments, which can lead to costly errors down the line. Regular meetings, discussions, and updates create an environment where team members can express concerns and provide feedback openly.

Lastly, collaboration can lead to increased productivity. When developers work together, they can divide tasks based on individual strengths and expertise. This division of labour enables the team to tackle multiple components of a project simultaneously, reducing development time and accelerating the delivery of the final product. Additionally, collaborative tools streamline workflows, allowing team members to track progress and manage tasks more efficiently.

A prime example of a collaboration tool used in software development is **GitHub**. GitHub is a platform that allows developers to host and review code, manage projects, and collaborate with others. It facilitates collaboration through features like pull requests, which enable team members to propose changes to the codebase and allow for peer review before merging those changes. This process not only enhances code quality through collective review but also provides a clear record of changes and discussions, fostering better communication among developers.

Programming for the web

Question 21

Marking Criteria	Marks
Accurately outlines the purpose of DNS, including its role in translating domain names to IP addresses for web data transmission.	2
Provides a partially correct or vague explanation of DNS or omits a key aspect of its purpose.	1

The Domain Name System (DNS) acts like the internet's phonebook. Its primary function is to translate human-readable domain names (like `www.example.com`) into machine-readable Internet Protocol (IP) addresses (like `144.6.2.1`). When you type a domain name into your browser, DNS lookups occur to find the corresponding IP address of the web server hosting that site, allowing your browser to connect to the correct server and retrieve the web page

Question 22

Marking Criteria	Marks
Clearly outlines the functions of Web Content Management Systems AND includes two examples.	3
Outlines the functions of Web Content Management Systems, with some lack of clarity OR provides a single example only.	2
Response shows some understanding of Web Content Management Systems OR provides a correct example.	1

Web Content Management Systems are software tools that simplify building and maintaining websites without needing to know how to code. Such systems include simple editors that are much like using a word processor—so you can add text, images, and videos without any technical skills. The design of the site stays consistent using built-in or purchased templates.

Different people work on a site with their own roles, like writers, editors and admins. You can schedule when content goes live, manage media files, and roll back to older versions if needed. Many systems also support plugins or add-ons that let you add features like contact forms, galleries, or online shops.

WordPress is perhaps the most popular example as it is readily available (many hosting plans include WordPress), beginner-friendly and has numerous included themes and plugins. Drupal is another more advanced system better for larger or complex sites, offering lots of flexibility for developers. All Web Content Management Systems aim to simplify managing a website.

Question 23

Marking Criteria	Marks
Clearly outlines the exchange between browser and web server including: • Confirm the identity of the web server • Establish a secure session • Communicate securely during the session	4
Outlines the exchange between browser and web server but with some lack of clarity or lack of accuracy	3
Outlines the exchange between browser and web server but neglects some significant or central aspects of the exchange	2
Response shows some understanding of a secure https session	1

To confirm the identity of the web server, the browser sends a request to connect using HTTPS. The server responds by sending back its SSL/TLS certificate, which includes its public key and is digitally signed by a trusted Certificate Authority (CA). The browser checks that the certificate is valid, matches the website's domain name, hasn't expired, and is issued by a trusted CA. If everything checks out, the browser trusts that it's talking to the legitimate server.

To establish a secure session the browser and server perform a handshake process. During this handshake, the browser generates a random "secret" value, encrypts it using the server's public key from the certificate, and sends it to the server. Only the server can decrypt this secret using its own private key. Both the browser and server then use this shared "secret" to generate a session key, which is used to encrypt and decrypt data for the rest of the communication session.

Once the secure session is established, the browser and server can communicate securely using the shared session key. This key is used to both encrypt and decrypt all messages. This ensures that all data sent between them can't be read or altered by third parties. This protects sensitive information like login details, personal data, and payment information as the user interacts with the website. The session remains secure until it is closed by either the browser or the server.

Question 24

Marking Criteria	Marks
• The response provides a comprehensive outline of key design principles for the PWA, addressing all specified aspects: font selection and typography, colour schemes and contrast, audio and video elements, and navigation design. • Each aspect is thoroughly explained with clear reasoning on how it enhances accessibility and inclusivity. • Includes relevant examples or best practices that illustrate the application of these principles. • Uses terminology that demonstrates a strong grasp of UI/UX concepts.	6
• The response effectively outlines key design principles for the PWA, covering all specified aspects with good detail. • Most aspects are well-explained, with clear connections to accessibility and inclusivity. • Provides at least one relevant example or best practice to support the discussion. • Generally uses correct terminology with minor inaccuracies.	5
• The response covers several key design principles, addressing most of the specified aspects but lacking depth in one or two areas. • Each aspect is explained but may have limited detail on how it contributes to accessibility and inclusivity.	4

- Includes at least one example or best practice - Uses mostly correct terminology with some inaccuracies.	
- The response mentions key design principles but lacks depth in explanation and detail. - Addresses only a few of the specified aspects, with minimal clarity on their relevance to accessibility and inclusivity. - Provides little to no examples or best practices to illustrate the concepts. - Uses basic terminology, demonstrating a limited understanding of UI/UX principles.	3
- The response provides a very brief mention of design principles with no clear structure or connection to the specified aspects. - Lacks detail and coherence, with unclear or vague explanations. - Does not include examples or best practices. - Uses limited terminology, demonstrating little understanding of UI/UX concepts.	2
- The provides basic information on the key design principles. - Lacks any meaningful structure or coherent explanation. - No examples or terminology used, showing a lack of understanding of the topic.	1

When designing a Progressive Web App (PWA) for infant school students, the focus must be on creating a friendly, accessible, and engaging experience that suits young learners' needs. The user interface (UI) should be simple, visual, and intuitive, while the user experience (UX) should support independence, exploration, and inclusivity for all users, including those with learning or physical disabilities.

- Font selection and typography

For early readers, fonts should be clear, rounded, and sans-serif, such as Comic Sans, Arial Rounded. Letter spacing and line height should be generous to improve readability, and text should be kept large and bold enough for developing eyes. Avoid using all caps and italics and consider supporting text-to-speech options for non-readers.

- Colour schemes and contrast

Colour choices should be bright and inviting, but also high in contrast to assist visually impaired users. For example, use dark text on a light background or vice versa, and follow WCAG (Web Content Accessibility Guidelines) for colour contrast ratios. Avoid relying on colour alone to convey meaning and provide options for dark mode or low-vision-friendly themes where needed.

- The use of audio and video elements

Audio and video should be used to support learning and engagement—such as interactive stories, spoken instructions, or songs—but must include controls to pause, stop, or replay. All audio content should have visual support (like subtitles or animations), and all video content should be captioned. Audio instructions should be clear, slow, and in a friendly tone suitable for young children.

- Navigation design

Navigation should be simple, icon-based, and placed consistently on each page. Use large, touch-friendly buttons with clear images and labels, and avoid deep menus or complex pathways. Include voice prompts or sound feedback for actions to guide non-readers. A "home" button should always be easily accessible, and progress made obvious.

Software automation

Question 25

Marking Criteria	Marks
Clear distinction between supervised and unsupervised learning	2
Response indicates a limited understanding of the distinction	1

Supervised learning involves training a model using labelled data, where each input is matched with a known output. The model learns the relationship between the input and output to make predictions. For instance, it could be used to estimate house prices based on factors like size or location.

Unsupervised learning is trained on unlabelled data, meaning no correct answers are provided. The model attempts to identify patterns or groupings within the dataset on its own. An example would be segmenting customers based on their shopping habits without predefined categories.

Question 26

Marking Criteria	Marks
Shows clear understanding of the use of weights to modify inputs to a neural network, the use of backpropagation to make adjustments based on the error score, and the need for many repetitions of this process in order to achieve accurate results.	3
Shows a good understanding of the use of weights to modify inputs to a neural network and the use of backpropagation to make adjustments.	2
Shows some understanding of how weights or backpropagation are used in a neural network	1

Weights in a neural network control how much influence each input has on the output. During training, the network randomly assigns weights, which are multiplied by the input, and compares the output to the correct answer. The system then calculates the error and, in a process called backpropagation, goes backwards through each layer in the network, adjusts the weights in each layer to reduce this error. Over many repetitions, the network improves its accuracy and learns the best weights for each input.

Question 27

Marking Criteria	Marks
Recommends a suitable ML regression algorithm, justifies the recommendation, clearly explains how it would be implemented using OOP, and lists two relevant features	4
Identifies an appropriate algorithm and either: – Outlines how it would be implemented using OOP, or – Lists two relevant model features	3
Identifies an ML regression algorithm and outlines some features or basic OOP steps	2
Provides some relevant information about regression, OOP, or model features	1

Appropriate ML algorithm:

The most appropriate algorithm is Linear Regression, as it predicts a continuous numeric value (test score) based on one or more input features (like hours studied).

Implementation using OOP:

Using an object-oriented approach, I would use a class-based model from a library like scikit-learn. I would:

- Import the LinearRegression class
- Instantiate the model: `model = LinearRegression ()`
- Use `.fit()` to train the model on input/output data
- Use `.predict()` to make predictions based on new data

Two features are required by the model:

- Hours studied (as the key input feature)
- Past test scores (as labelled output to train the model)

Question 28

Marking Criteria	Marks
Produces a substantially correct algorithm which: • Accepts the correct parameters and returns the label • Calculates distance between the test point and all training points • Selects the k closest points • Extracts labels and determines the most common label • Uses provided helper functions appropriately and efficiently	6
Produces an algorithm which completes at least four of the listed steps with minor errors.	5
Produces an algorithm which completes at least three of the listed steps but contains errors.	4
Produces an algorithm which completes at least two of the listed steps but contains errors.	3
Produces an algorithm which attempts some of the listed steps, but with significant errors.	2
Makes an attempt at an algorithm which relates to the question.	1

```

BEGIN knn(newPoint, tData, tLabels, k)
    distances = []
    FOR i = 0 TO length of tData STEP 1
        d = distance(newPoint, tData[i])
        append [d, tLabels[i]] to distances
    NEXT i
    distances = sort(distances)

    nearest = []
    FOR i = 0 TO k - 1 STEP 1
        append distances[i][1] to nearest
    NEXT i

    label = mode(nearest)
    RETURN label
END

```

The above pseudocode uses Python-like syntax to specify arrays as lists. There is no direction in course specifications, so given Python is now specified we presume such syntax is preferred or at least will be equally acceptable to HSC examiners.

Software engineering project

Question 29

Marking Criteria	Marks
Clearly explains multiple ways Gantt charts assist project managers to meet deadlines.	2
Response indicates understanding of Gantt charts with some relevance to deadlines.	1

Gantt charts help project managers by showing all the tasks and when they need to be completed, making it easier to plan and meet the 12-month deadline. They also display task dependencies, so managers can identify which tasks must finish before others start. Finally, Gantt charts allow tracking progress regularly, helping to spot delays early and keep the project on schedule.

Question 30

Marking Criteria	Marks
Algorithm correctly mirrors the structure chart, including order, decision and all parameters.	3
Algorithm mirrors the structure chart to some extent but misses either order, decision or all parameters.	2
Response demonstrates some understanding of the relationship between structure charts and algorithms.	1

```
BEGIN FindItem(ItemID)
    Found = ItemExists(ItemID)
    IF Found THEN
        Location = CheckLocation(ItemID)
    ENDIF
    RETURN Location
END
```

Question 31

Marking Criteria	Marks
Excellent discussion the role and importance of programmed backups and the role and importance of version control.	4
Thorough discussion the role and importance of programmed backups and the role and importance of version control lack of clarity.	3
Discusses the role and importance of programmed backups OR the role and importance of version control lack of clarity.	2
Response shows some understanding of backups OR version control.	1

When producing and implementing code, programmed data backup ensures that project files, databases, and environments are regularly saved and can be recovered in case of accidental loss, system failure, or corruption. Automated backup systems reduce the risk of losing work, especially during major updates or modifications, and are essential for maintaining data integrity throughout the development lifecycle.

On the other hand, version control systems like Git are used to track changes in the codebase, allowing developers to work collaboratively, manage updates, and revert to earlier versions if needed. Version control records the history of changes, making it easier to identify when and where bugs were introduced and to coordinate teamwork across multiple programmers.

During code implementation, combining both tools provides a safety net: version control manages the logical flow and collaboration of coding, while backups protect the broader system—including files not typically tracked in version control, like databases and configuration settings. This ensures recovery is possible even if a deployment error affects the live environment.

Question 32

Marking Criteria	Marks
• Clear, accurate description of both Wagile (hybrid planning + iteration) and Agile (fully iterative, feedback-driven). • One valid advantage and disadvantage for each approach. • Specific scenario with logical justification for why Agile suits it better (e.g., changing requirements in startups).	6
• Correct description of both Wagile (hybrid planning + iteration) and Agile (fully iterative, feedback-driven). • One valid advantage and disadvantage for each approach. • Provides a satisfactory scenario and attempts to justify why Agile suits it better.	5
• Mostly correct description of both Wagile (hybrid planning + iteration) and Agile (fully iterative, feedback-driven), AND • Some valid advantages and disadvantages for both approaches, AND • Provides a satisfactory scenario and attempts to justify why Agile suits it better.	4
• Partially correct description of both Wagile and Agile, AND • Some valid advantages and disadvantages for both approaches, OR • Provides a satisfactory scenario and attempts to justify why Agile suits it better.	3
Any <u>two</u> of the following: • Identifies only advantages or disadvantages, or covers just one approach, OR • Describes only one approach correctly, OR • Identifies a scenario but lacks justification	2
Any <u>one</u> of the following: • Identifies only advantages or disadvantages, or covers just one methodology, OR • Describes only one methodology correctly, OR • Identifies a scenario but lacks justification	1

Wagile blends Waterfall's structured planning with Agile's iterative delivery. It begins with upfront requirements gathering (like Waterfall) but implements features in sprints (like Agile), allowing some flexibility while maintaining project milestones.

Pure Agile is fully iterative, focusing on delivering working software in short cycles through continuous customer collaboration. It prioritises responding to change over following a fixed plan, with less upfront documentation.

Wagile's advantage: balances predictability (from Waterfall) with adaptability (from Agile), suitable for projects needing clear timelines but some flexibility. Wagile's disadvantage: can be inefficient if initial plans require frequent revisions, leading to delays.

Agile's advantage: excels in dynamic environments due to rapid response to feedback. Agile's disadvantage: risk of scope creep without rigid deadlines, potentially increasing costs.

One scenario where pure Agile would be more suitable: A mobile app startup developing a fitness tracker. With the Agile approach, short sprint reviews (e.g. weekly) to incorporate user feedback on features like heart-rate monitoring. However, Wagile's upfront design phase might delay these adjustments. For example, if beta testers request a sleep-tracking feature, Agile teams can add it in the next sprint, while Wagile would require revisiting initial plans.

Mapping Grid

Question	Marks	Content	Syllabus outcomes	Targeted performance bands
1	1	Secure software architecture	SE-12-01	2-3
2	1	Secure software architecture	SE-12-03	3-4
3	1	Programming for the web	SE-12-03	3-4
4	1	Programming for the web	SE-12-06	3-4
5	1	Software automation	SE-12-05	4-5
6	1	Software automation	SE-12-06	4-5
7	1	Software engineering project	SE-12-01	3-4
8	1	Software engineering project	SE-12-01	3-4
9	1	Secure software architecture	SE-12-01	3-4
10	1	Programming for the web	SE-12-08	3-5
11	1	Software automation	SE-12-05	3-5
12	1	Software engineering project	SE-12-09	3-5
13	2	Secure software architecture	SE-12-06	4-5
14	2	Programming for the web	SE-12-05	3-5
15	2	Software automation	SE-12-06	4-6
16	2	Software engineering project	SE-12-07	2-4
17	2	Secure software architecture	SE-12-01	3-4
18	3	Secure software architecture	SE-12-04	3-5
19	4	Secure software architecture	SE-12-01 SE-12-04	4-6
20	6	Secure software architecture	SE-12-01 SE-12-05	3-6
21	2	Programming for the web	SE-12-03	2-4
22	3	Programming for the web	SE-12-03	2-4
23	4	Programming for the web	SE-12-03	4-6
24	6	Programming for the web	SE-12-06	3-5
25	2	Software automation	SE-12-03	3-5
26	3	Software automation	SE-12-06	4-6
27	4	Software automation	SE-12-06	3-5
28	6	Software automation	SE-12-07	3-6
29	2	Software engineering project	SE-12-09	2-4
30	3	Software engineering project	SE-12-02	4-6
31	4	Software engineering project	SE-12-09	3-5
32	6	Software engineering project	SE-12-09	2-5