DO NOT WRITE ON THIS EXAM PAPER, USE THE ANSWER SCRIPT PROVIDED.

This paper consists of two sections: **SECTION A** and **SECTION B**.

You must answer **ALL** questions in **SECTION A** and **TWO** questions from **SECTION B**.

SECTION A is worth **40 marks** in total, while each question in **SECTION B** is worth **30 marks**. The marks for each part of a question are shown in [ ] brackets at the end of the question.

SECTION A (Multiple choice questions, answer all questions in this section)

Question 1 (Multiple Choice, please write the letter of the part and the number of the correct answer in your booklet)

a) In the context of Interaction Design, which term refers to the logical relationship between controls and their effects (e.g., pressing the 'up' arrow moves the page up)?

1. Affordance
2. Mapping
3. Constraint
4. Feedback
5. Signifier [4 marks]

b) Which of the following is considered a "Code Smell" that might indicate a violation of the **Don't Repeat Yourself (DRY)** principle?

1. A class with a single responsibility.
2. Duplicated logic blocks copied across multiple functions.
3. Using dependency injection to pass objects.
4. Writing unit tests before the actual code.
5. Using descriptive variable names. [4 marks]

c) What is the specific function of the git blame command in a version control workflow?

1. It deletes the last commit from the history permanently.
2. It shows who last modified each line of a file and when.
3. It resolves merge conflicts automatically by accepting "theirs".
4. It creates a new branch based on a specific tag.
5. It pushes changes to the remote repository forcefully. [4 marks]

d) According to security best practices, what is the primary advantage of using **Multi-Factor Authentication (MFA)**?

1. It speeds up the login process for the user.
2. It encrypts the database using AES-256 automatically.
3. It blocks approximately 99.9% of account compromise attacks.

4. It removes the need for the user to remember a password.
5. It prevents SQL Injection attacks on the login form. [4 marks]

e) Which usability testing method involves asking users to organize information into groups to help design the Information Architecture (navigation)?

1. Heuristic Evaluation
2. A/B Testing
3. Card Sorting
4. Eye Tracking
5. Load Testing [4 marks]

f) In the context of performance profiling, what is the purpose of tools like `htop` or `perf`?

1. To validate HTML syntax against W3C standards.
2. To manage Git merge conflicts visually.
3. To identify system bottlenecks and optimize code performance.
4. To automatically write unit tests for Python functions.
5. To scan the code for security vulnerabilities like XSS. [4 marks]

g) What is the "Staging Area" in Git?

1. A remote server where code is deployed for beta testing.
2. A temporary holding area for changes before they are committed.
3. A separate branch used only for experimental features.
4. The local directory where the `.git` folder is hidden.
5. A backup drive where repositories are cloned. [4 marks]

h) Which of the following best describes the **Interface Segregation Principle (ISP)**?

1. High-level modules should not depend on low-level modules.
2. Classes should have only one reason to change.
3. Clients should not be forced to depend on interfaces they do not use.
4. Derived classes must be substitutable for their base classes.
5. Software entities should be open for extension but closed for modification. [4 marks]

i) Ideally, what should happen to the "Technical Debt" of a project when **Refactoring** is applied correctly?

1. It should increase significantly.
2. It should remain exactly the same.
3. It should be converted into financial debt.
4. It should be reduced, leading to improved maintainability.
5. It should be hidden from the project manager. [4 marks]

j) Which type of coupling is considered the **worst** (tightest) form, where one module directly modifies the internal data of another?

1. Data Coupling
2. Message Coupling

3. Stamp Coupling
4. Control Coupling
5. Content Coupling [4 marks]

SECTION B (Written questions, this section contains three questions, each worth 30 marks. You must select and answer ONLY TWO of them.)

Question 2: Secure Deployment and Memory Management (30 marks)

a. Supply Chain Vulnerabilities: The "SolarWinds Hack" and "Log4j Vulnerability" highlighted major risks in modern software development. i. Explain what a **Supply Chain Attack** is. Why is securing third-party dependencies often more difficult than securing your own code? ii. Describe one lesson learned from the **Equifax Data Breach** regarding patch management. [10 marks]

b. Memory Safety: In languages like C++, memory management is manual. i. Explain the concept of a **Memory Leak**. How does it occur and what is the long-term impact on a running server? ii. Name one tool mentioned in the lecture material that can help identify memory safety errors. [5 marks]

c. Secure Configuration & Containers: Misconfigurations are a leading cause of cloud breaches. i. Explain why **Containerization** (using tools like Docker) is preferred over manual deployment for ensuring security and consistency across different environments. ii. Referencing the **Capital One** breach, explain the danger of a misconfigured firewall or excessive access privileges. [5 marks]

d. Encryption: i. Describe **AES (Advanced Encryption Standard)** using the "Secret Blender" analogy. Why is the configuration of AES considered secure against brute-force attacks? ii. Explain the difference between **Authentication** and **Authorization**. [10 marks]

Question 3: Usability Engineering and User Research (30 marks)

a. Information Architecture: i. Describe the **Card Sorting** method (Closed vs. Open). How does this specific activity help designers create intuitive navigation structures (menus)? ii. Contrast this with **User Journey Mapping**. What different aspect of the user experience does a Journey Map visualize compared to a Card Sort? [10 marks]

b. Profiling and Performance: i. Define **Eye-Tracking Studies** and **Heatmaps**. How can these visual tools reveal flaws in a UI's visual hierarchy that a standard survey might miss? ii. Explain the significance of the **System Usability Scale (SUS)**. What score is generally considered the threshold for "above average" usability? [10 marks]

c. Accessibility (Inclusive Design): i. Explain the concept of **Perceivable** information under WCAG guidelines. Give an example of how "Alt Text" for images satisfies this principle. ii. Beyond legal compliance, explain the **business argument** for Accessibility. How does it affect the potential market size and user engagement? [10 marks]

Question 4: Design Principles and Git Internals (30 marks)

a. DRY (Don't Repeat Yourself): i. Analyze the following scenario: A developer copies a SQL query string into five different functions, changing only the `ID` variable in each. Identify the maintenance risk if the database schema changes (e.g., a column is renamed). ii. Propose a specific refactoring technique (e.g., using a Template or centralized function) to resolve this redundancy. [5 marks]

b. Coupling and Cohesion: i. Define **Content Coupling**. Why is this considered the "worst" form of coupling compared to **Data Coupling**? ii. Explain the relationship between **High Cohesion** and **Modular Design**. Why should a module be organized around a "single, well-defined purpose"? [10 marks]

c. Git Internals: i. Explain the purpose of the **Staging Area** (Index) in Git. Why does Git require a two-step process (`git add` then `git commit`) rather than just saving the file state directly? ii. Describe the function of the `.gitignore` file. Why is it critical for both repository cleanliness and security (e.g., avoiding the commit of API keys)? [5 marks]

d. Code Reviews: i. Beyond simply finding bugs, explain two "soft" benefits of conducting **Peer Code Reviews** for the development team (e.g., knowledge sharing). ii. What is the role of a **Checklist** in a code review? Name one item that should be on a standard code review checklist regarding **Readability** or **Security**. [10 marks]