PYQ June 2025

Question 1

Question

a) Examine the primary uses of the following types of scripting languages, along with **ONE (1)** example of a specific language for each type

(i) Shell Scripting Languages *(3 marks)*
(ii) Extension Languages *(3 marks)*
(iii) Web Scripting Languages *(3 marks)*

Answer

(i)

- Shell Scripting Languages automates and glue together the operating system commands to execute the system administration, batch processing and task automation.
- It contains the sequence of commands that are executed by the operating system's command interpreter (shell).
- For example, PowerShell and Bash.

(ii)

- Extension Languages are used to control, customize or extend the functionalities of the existing system.
- These scripts do not run independently but are embedded inside software to add automation and customization features.
- For example, JavaScript and VBScript.

(iii)

- Web Scripting Languages are used to develop a dynamic web pages and handle the interactions between users and web applications using client-side and server-side scripting.
- For example, JavaScript, Python and PHP.

Question

b) You are designing a secure authentication system for a banking application. The system must ensure the functionality, integrity and availability of user data. Examine **TWO (2)** security best practices you would implement to secure the authentication process. *(4 marks)*

Answer

- During login process of the banking application, the system must implement a strong cryptographic hashing algorithms for storing the user passwords instead of plaintext storage. For example, bcrypt or Argon2. This can effectively prevent the credentials from being unauthorized accessed by attackers or recovered even if the database is compromised.
- Multi-Factor Authentication (MFA) should be applied when login the banking application or executing sensitive actions like transferring money which exceeds RM1000. For example, fingerprint, One-Time Password (OTP) via email or SMS or face recognition and verification. Thus, the attackers would not be able to access the user's bank account and transfer the money even though they has obtained the user's password.

Question

c) You are developing a web application that allows users to log in and submit sensitive information *(e.g., credit card details)*.

(i) Identify **THREE(3)** security vulnerabilities that could exist in this application and explain how they could be exploited. *(3 + 6 marks)*
(ii) For each vulnerability that you identified, list **ONE(1)** mitigation strategy to prevent or reduce the risk. *(3 marks)*

Answer

(i)

- SQL injection
  - When filling in the login form, the attackers may inject the malicious SQL code inside the field and submit the form.

    - This action may trick the system to execute the malicious SQL code in the query and seriously affect the database if there is any mechanism taken to filter or normalize the input.
    - Via this security vulnerability, the attackers can easily bypass the authentication and authorization process and directly access to the system accounts or even destroy and modify the system database.
- Cross-Site Scripting (XSS)
    - The attackers may inject the malicious JavaScript script into the web pages and let the users accidentally execute it and perform the unintended actions which can help the attackers.
    - The attackers may create a request and inject into the web application as a button or hyperlink or even a fake form.
    - When the users accidentally click on it, it will navigate the users to the intended website and steal the users' cookie session data if the users have logged in to the web application.
    - Eventually, the attackers can access to the web application with the victim's user account using the stolen cookie session data.
- Denial of Service (DoS)
    - The attackers may try to make the system, network or services provided by the web application unavailable by overwhelming it with numerous request traffic.
    - When there are too many pending request traffic sent to one server until exceeding the acceptable workload by the server, the server will be down automatically.
    - Eventually, all the services will be down and the normal users will not be able to access the services.

(ii)

- SQL injection - Prepared statement or parameter binding in SQL statement
    - When the system want to do any interaction with the database especially when involving external user input, the system should use prepared statement.
    - The prepared statement will definitely separate the query and the data they need to bind with it,
    - All the user input will be automatically treated as a plain text form instead of the executable command.
- Cross-Site Scripting (XSS) - Input sanitization
    - When receiving the incoming data, the system should apply the input sanitization which can remove or normalize the dangerous or malicious characters.

    - Thus, the data will only be treated as a plain text and would never be executed by the system or browser.
    - For example, `<script>` will be normalized into `&lt;script&gt;`.
- Denial of Service (DoS) - Traffic filtering and rate limiting
    - The web application server should implement the traffic filtering mechanism to prevent all the malicious traffic from reaching to the server by dropping or discarding the traffics.
    - This can effectively filter all the malicious request made by attackers.
    - The rate limiting should also be applied to establish restriction on the frequency of the request can be made within a certain timeframe. For example, only 20 requests can be made from an IP address within 5 minutes.
    - This can prevent a lot of intended traffic overwhelming actions and maintaining the smooth operation of the server without reaching the maximum point.

Question 2

Question

a) Examine **ONE(1)** difference between the **Adapter Pattern** and the **Facade Pattern**. Provide a scenario where each would be more appropriate to use. *(6 + 4 marks)*

Answer

- The Adapter Pattern is used to allow two classes with incompatible interfaces to work together by converting the interface of an existing class into one expected by the client. Meanwhile, the Facade Pattern is used to provide a simplified and unified interface to a complex subsystem and hide the internal complexity from the client.
- The Adapter Pattern is appropriate when a system needs to reuse and existing or legacy class whose interface does not match the required interface without modifying the original class.
- The Facade Pattern is appropriate when a system consists of multiple complex subsystem classes and a single, simplified interface is needed to make the system easier for clients to use and reduce coupling.

Question

b) A fitness tracking application allows uses to calculate their daily calorie burn based on different types of physical activities, such as Running, Cycling, and Swimming. Each activity has a unique formula for calculating calories burned. The application should allow users to switch between activities dynamically and should be easily extendable to support new activities in the future.

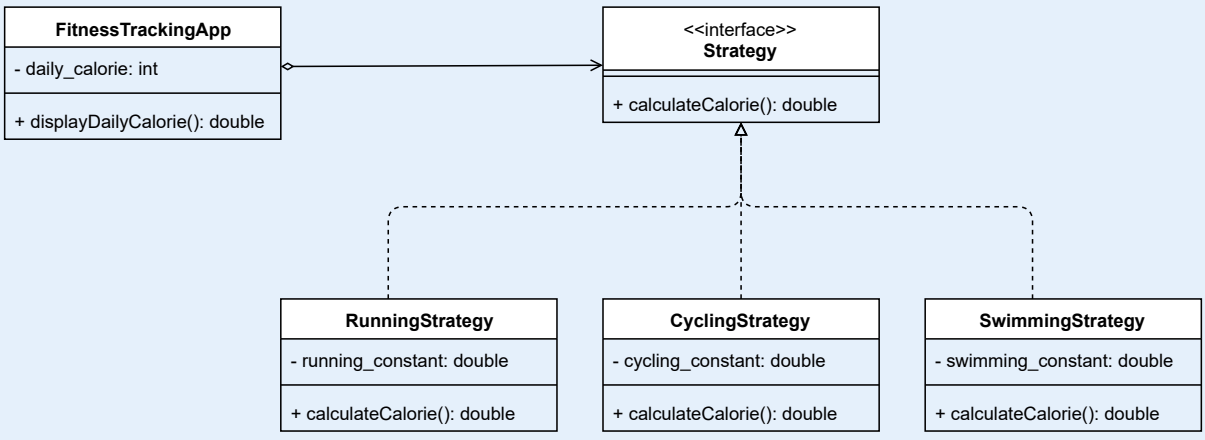
(i) Identify the most suitable design pattern for this scenario. Justify your choice. *(5 marks)*
(ii) Draw a class diagram to illustrate how the design pattern would be implemented in this system. Ensure your diagram includes all necessary classes, interfaces and relationships. *(10 marks)*

Answer

(i)

- Strategy design pattern.
- Different algorithms will be taken to calculate the users' daily calories burn based on different physical activities chosen.
- The Strategy design pattern can easily handle this situation by delegating the calculation to different concrete strategy classes at runtime based on the user selected physical activities without modifying the existing context classes. Thus, new activities can be added without changing the existing code.
- For example, the concrete strategy class `RunningStrategy` will be accessed when the user has chosen the `Running` activities. Within the the `RunningStrategy` class, the formula and calorie burn calculation method which is specifically designed for Running activities would be provided.

(ii)



Question 3

Question

a) You are tasked with creating an eXtensible Markup Language (XML) file to store information about books in a library. The file must support international characters (e.g. accented characters, Chinese, or Arabic text) and special symbols.
Figure 1 is an example XML document, but the encoding rule has not been specified in the XML declaration. The `id` is a required attribute, and the `availability` element is restricted to one of three values: "`Available`", "`Checked Out`" or "`On Hold`".

(ii) Examine the suitable encoding rule to be used in this scenario. Provide **ONE (1)** reason for your choice. *(4 marks)*
(ii) Write a Document Type Definition (DTD) to validate the structure of the XML document. *(9 marks)*
(iii) Analyse **TWO (2)** limitations of using DTD for this purpose. *(4 marks)*

```xml
<?xml version="1.0" encoding="_____"?>
<library>
    <book id="bk001">
        <title>The Great Gatsby</title>
        <author>Jose Lopez</author>
        <genre>Classic</genre>
        <publication_year>1925</publication_year>
        <isbn>9780743273565</isbn>
        <availability status="Available">In stock
        </availability>
    </book>
    <book id="bk002">
        <title>To Kill a Mockingbird</title>
        <author>Harper Lee</author>
        <genre>Fiction</genre>
        <publication_year>1960</publication_year>
        <isbn>9780061120084</isbn>
        <availability status="Checked Out">Due back in 7 days
        </availability>
    </book>
</library>
```

Answer

(i)

- UTF-8 is the most suitable encoding rule for this XML document.
- It can support all the Unicode characters and allow the XML file to store the international characters such as accented letters, Chinese, and Arabic text as well as special symbols.
- It is also backward-compatible with ASCII to make it widely supported and suitable for cross-platform XML data storage.

(ii)

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT library (book+)>
<!ELEMENT book (title, author, genre, publication_year, isbn,
availability)>
<!ATTLIST book id ID #REQUIRED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT publication_year (#PCDATA)>
<!ELEMENT isbn (#PCDATA)>
<!ELMEENT availability (#PCDATA)>
<!ATTLIST availability status (Available | Checked Out | On Hold)
#REQUIRED>
```

(iii)

- DTD only supports few data types like CDATA, ID and IDREF, it cannot define the numeric ranges, dates or custom data types like integer, string and Boolean.
- DTD does not support XML namespaces.
- DTD uses a different syntax from XML, it will require the developers to learn additional syntax.
- DTD cannot enforce constraints like value ranges, pattern matching or enumeration with strong typing.

Question

b) You are developing a financial analytics application that processes XML files containing market data. The application needs to handle two types of XML files:

- **File A**: A small XML file (a few kilobytes in size) that contains metadata about stock symbols (e.g., company name, sector, and industry). This file is used to populate a dropdown menu in the application's user interface.

- **File B**: A large XML file (several gigabytes in size) that contains real-time stock price updates (e.g., timestamp, stock symbol, and price). The application needs to extract specific stock prices from this file as quickly as possible.

For each file, examine whether you would use Simple API for XML (SAX) or Document Object Model (DOM) parsing. Provide detailed reasoning for your choices. *(8 marks)*

Answer

- File A
  - Document Object Model (DOM) parser is suitable for File A.
  - Since the XML file is small, loading the entire document into memory does not cause performance or memory issues.
  - DOM converts the XML document into a tree structure, allowing easy navigation and repeated access to elements such as company name, sector and industry.
  - This makes it ideal for populating and reusing data in a user interface such as a dropdown menu.
- File B
  - Simple API for XML (SAX) parsing is suitable for File B.
  - The XML file is very large, and building a DOM tree would consume excessive memory and reduce performance.
  - SAX processes the XML document sequentially and triggers events when relevant elements are encountered, allowing the application to extract only the required stock price data.
  - This makes SAX more memory-efficient and faster for processing large or streaming XML files.

Question 4

Question

a) You are designing a web service for an online bookstore. The service allows clients to search for books, view book details, and place orders. The service must support both Simple Object Access Protocol (SOAP)-based and RESTful architectures.

(i) Compare SOAP and RESTful web services by outlining **THREE (3)** key differences between them. *(9 marks)*
(ii) Examine **TWO (2)** advantages of using JavaScript Object Notation (JSON) over eXtensible Markup Language (XML) for data exchange in RESTful web services. *(4*

marks)
(iii) Outline **ONE (1)** scenario where SOAP would be a better choice than REST for implementing the bookstore service. Justify your answer. *(4 marks)*

Answer

(i)

- SOAP web service is a protocol that allows for exchanging the data from one system to another system using XML message format and must be strictly followed while the RESTful web service is similar to SOAP web service but it is an architectural style recommendation which does not need to be strictly followed.
- SOAP web service only supports XML format which uses opening and closing tags while RESTful web service can support both XML and JSON (most commonly used).
- SOAP web service uses HTTP mainly as a transport mechanism while RESTful web service will apply HTTP as part of its design which uses the HTTP request method like GET, POST, PUT, DELETE or PATCH and status code like 200, 401, 404 and 500.
- SOAP web service has built-in security mechanisms such as WS-Security, encryption and authentication while RESTful web service will rely on external security mechanism like HTTPS and token-based authentication.

(ii)

- JavaScript Object Notation (JSON) can support various data type like objects, arrays, strings, numbers and so on while eXtensible Markup Language (XML) can only support element, attributes and text.
- JSON is more lightweight and less verbose than XML so it results a smaller message sizes and faster parsing, which improves the performance in RESTful web services.

(iii)

- During the payment session for placing orders, SOAP will be a better choice than REST.
- This is because SOAP has a well-defined and structured security mechanisms including encryption, authentication and so on. This can effectively secure the clients' payment credentials when making the payment.
- Meanwhile, by default, SOAP also strictly enforces the message validation, XML format and code best practices to pass the request message between systems. This can prevent any threat or risk of request message manipulation or modification when sending the request to each other in a more advanced way.

- For REST, the security is not defined as part of the architecture itself and is typically implemented using external mechanisms such as HTTPS, OAuth or token-based authentication. In contrast, SOAP provides standardized, built-in security specifications such as WS-Security, which define how security features like encryption, authentication and message integrity should be implemented.

Question

b) You are tasked with developing a client-server application for a chat system. The system allows multiple clients to connect to a server and exchange messages in real-time. The server must handle multiple client connections simultaneously and broadcast messages from one client to all others.

(i) Analyse the key difference between Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) in the context of network communication, focusing on their reliability, and connection handling. *(5 marks)*
(ii) Evaluate which protocol (TCP or UDP) would be more suitable for the chat system, justifying your choice based on the system's requirements. *(3 marks)*

Answer

(i)

- TCP is a connection-oriented protocol that ensures reliable data transmission through acknowledgements, sequencing and retransmission of lost packets.
- It guarantees ordered and error-free delivery of messages.
- In contrast, UDP is a connectionless protocol that does not guarantee delivery, ordering or reliability, but offers faster transmission with lower overhead.

(ii)

- TCP would be more suitable for the chat system.
- It is because the chat messages must be delivered reliably and in the correct order.
- TCP ensures that no messages are lost or duplicated, which is important for real-time communication between clients.

- Although TCP introduces some overhead, its reliability makes it appropriate for a chat application.