H2 COMPUTING (9597)

DEFINITIONS AND QUALITATIVE ANSWERS BANK

1. Algorithms and Data Structures

1.1 Recursive Procedure / Function

A *recursive procedure* is a program that calls itself with smaller and smaller subproblems till it reaches a terminating base case which can be solved directly, then backtracks and solve the problem.

Give the two line numbers which indicate that the procedure is recursive. [2016]

- Line that shows a **terminating base case** – `IF T[INDEX] > 0`
- Line that shows recursive calls with smaller and smaller subproblems – `X(Value, Index * 2)`

1.2 BST vs. Hashing

- We can get all keys in sorted order by just doing Inorder Traversal of BST. This is not a natural operation in Hash
- Tables and requires extra efforts.
- BSTs are easy to implement compared to hashing, we can easily implement our own customized BST. To implement Hashing, we generally rely on libraries provided by programming languages.
- With Self-Balancing BSTs, all operations are guaranteed to work in $O(logn)$ time. But with Hashing, $O(1)$ is average time and some particular operations may be costly, especially when table resizing happens.

1.3 Stack and Queue Applications

How does a printer receive and control print jobs?

Every print job is numbered and has an ID. There is also a print job *queue* present in the printer or another hardware to manage the order of print jobs. The first print job request is processed first, and the last print job request is processed last in the queue structure. The printer can initiate connection to the terminals according to the printer queue so that the documents to be printed can be spooled for printing.

How can $(A + B) \times (C + D)$ be calculated using a stack data structure? [2017]

1. Convert from Infix notation to postfix notation e.g. 3 + 4 to 3 4 + using a stack and a queue

2. Evaluate the postfix notation using a stack. Steps involved:
   A) Push operands into stack
   B) Every time an operator is encountered, Pop 2 operands from stack and perform the required mathematical operation.
   C) Push the result back to stack.

3. Repeat this until the end of postfix expression, result will be in the top item of the stack.

2. Object-Oriented Programming

2.1 Encapsulation

Encapsulation refers to the bundling of private data and public methods within a class. Private data may only be accessed by public methods, which prevents external code from interfering/modifying data values in the class.

Explain, using this example, why encapsulation is an example of good programming practice. [SP9597]

Encapsulation achieves 2 main objectives: (1) hiding complexity and (2) hiding the sources of change, which promote *abstraction* and *modularity* respectively.

Abstraction: It hides unnecessary/complex details from the users. It makes objects simpler to use and understand by defining strict external interfaces.

e.g. clients can always expect the `GetFuelPercentage()` method to return the fuel percentage of the car without being concerned with unnecessary details such as `FuelType` and `VehicleModel` as these attributes are irrelevant to the client needs.

Modularity: It decouples the modules that comprise a system, allowing them to be developed, tested, optimized, used, understood, and modified in isolation as long as the behaviour of the methods do not change. This speeds up system development because modules can be developed in parallel. It eases the burden of maintenance because modules can be understood more quickly and debugged with little fear of harming other modules.

e.g. If the programmer decides that an additional restriction is to be added, that the fuel percentage must be between 0 and 100 and must be a float, an additional validation can be added to the `SetFuelPercentage(NewPercentage)` method and the `FuelPercentage` attribute can be initialised as a float, instead of an integer. This can be done independently without affecting other modules because they do not depend on these details; they only depend on the behaviour of the `SetFuelPercentage()` method.

2.2 Inheritance

Inheritance refers to the ability to create subclasses which adspt all attributes and methods from an existing super class. This promotes code reusability.

For example, `FoodDelivery` is a subclass of `Good`, which inherits all attributes and methods. The subclass can have additional properties and methods implemented on top of its super class structure.

Inheritance is an important feature of OOP as it promotes code reusability as the super class is able to define common functionalities, while the sub-classes can extend them by adding new functionalities or override the existing ones.

2.3 Polymorphism

Polymorphism refers to the ability of different classes to respond to the same methods in different ways.
For example, `display()` method in different classes may function differently to meet different requirements.

2.4 Class

In object-oriented programming, a *class* is an extensible program-code-template for creating objects, providing initial values for state (member variables) and implementations of behaviour (member functions or methods).

2.5 Object

An *object* is an instance of a class at program runtime. Each object has its own state space but shares the same method space as other objects of the same class.

2.6 Instance

In object-oriented programming (OOP), an *instance* is a concrete occurrence of any object, existing usually during the runtime of a computer program.

Class → Object → Instance

A blueprint for a house design is like a class description. All the houses built from that blueprint are objects of that class. A given house is an instance.

```
K3 = Car("KIA", "2019", 98000, "15.1 km/L", ["Electric Seat", "Keyless engine start"])
```

In this case, the Car is a class describing states and behaviours of cars. K3 is an instance of the class Car, because it contains specific information of 1 car.

3. Data Validation and Verification

Explain the difference between data validation and data verification. [SP9597]

Validation is the **automated checking of data** to ensure data values are **acceptable / reasonable**.
Verification is process of ensuring **transferred data matches source data**

4. Testing and Debugging

4.1 Syntax Error

Syntax error is a structural error of a program such that the code violates grammar/rules of the programming language.

4.2 Semantic Error

Semantic error is due to wrong interpretation of the program statement.

e.g. `Week` is declared to be an integer yet a string `"0"` is assigned to it. (`Week ← "0"` does not violate the grammar of the programming language, but

4.3 Logic Error

Logic error is one which allows a program to run successfully, but produces an unintended or undesired result.

e.g. `WHILE DAYS != 0 DO DAYS = DAYS - 7` may cause an infinite loop if it is not a multiple of 7.

4.4 Debugging Techniques

Trace using print statements to output values of variables at suitable junctures of program (e.g. before/after loop and before/after function call / expression) – able to detect if variables contain correct values to meet criteria for subsequent conditional/loop controls.

Programmer can *set breakpoints and use the stack trace function* to inspect if functions are called with the correct parameters. A stack trace allows tracking the sequence of nested functions called, up to the point where the stack trace is generated. Hence, the programmer is able to detect wrong function calls or parameter values.

Programmer can use the *step-over and step-into features* to determine line by line execution of code within the calling and called functions (step-over if called function code need not be traced, step-into if called function code needs to be traced).

4.5 White Box Testing

White box testing, also known as *structural testing*, uses specific knowledge of program code to test the internal structure of the program and examine the outputs.

The internal structure of the item being tested is **known** to the tester.

The test data is carefully chosen to achieve maximum code coverage. The actual results are then compared to the expected results.

4.6 Black Box Testing

Black box testing, also known as *behavioural testing*, is used to check that the output of a program, given certain inputs, would conform to the functional specification of the program.

The internal structure of the item being tested is **not known** to the tester.

The test data is carefully chosen to incorporate a wide range of possible inputs. The actual results are then compared to the expected results.

Explain three differences between black box and white box testing. [2017]

	Black Box Testing	White Box Testing
Definition	*Black box testing*, also known as *behavioural testing*, is used to check that the output of a program, given certain inputs, would conform to the functional specification of the program. The internal structure of the item being tested is **not known** to the tester.	*White box testing*, also known as *structural testing*, uses specific knowledge of program code to test the internal structure of the program and examine the outputs. The internal structure of the item being tested is **known** to the tester.
Levels Applicable To	Mainly applicable to higher levels of testing: System Testing Acceptance Testing	Mainly applicable to lower levels of testing: Unit Testing Integration Testing
Responsibility	QA Team	Software Developers
Programming Knowledge	Not Required	Required
Implementation Knowledge	Not Required	Required
Basis for Test Cases	Derived from Requirement Specifications	Derived from Detail Design / Program Logic

	Tests **functionality** of an application. The test data is carefully chosen to incorporate a wide range of possible inputs.	Tests **internal structures** or workings of an application. The test data is carefully chosen to achieve maximum code coverage.

4.7 Top Down Testing

The highest level components are tested first and then step by step start working downwards (lower components)

4.8 Bottom-Up Testing

This is the opposite of top-down where the lower level components are tested first before slowing testing upward to the higher level components

4.9 Test Data

Explain, with examples, the different types of test data that could be used in a full test programme. [SP9597]

Normal data – data that should be accepted and normal processing performed e.g. `50`
Extreme/boundary data – data that tests against boundary/loop conditions e.g. `0, 60` (for the line `IF X >= 0 and X <= 60 THEN`)
Erroneous data – different data type e.g. `'abc'`, out of range e.g. `-1, 61`

5. Database

5.1 Data Inconsistency

Data inconsistency exists when different and conflicting versions of the same data appear in different places.

5.2 Data Redundancy

Data redundancy refers to the same data being stored more than once.

5.3 Data Dependency

Data dependency refers to information being stored in specific columns and programmer needing knowledge of format of data stored in files. RDBMS contains column headers to identify data. Hence it is not dependent on the file structure.

5.4 Candidate Key

A *candidate key* is defined as a minimal set of fields which can uniquely identify each record in a table. A *candidate key* should never be NULL or empty. There can be more than one *candidate key* for a table.

5.5 Primary Key

A *primary key* is the minimal set of fields which can uniquely identify each record in a table, that is most appropriate to become the main key for a table. A *primary key* should never be NULL or empty. There is only one *primary key* for each table.

OR

A *primary key* is a candidate key that is most appropriate to become the main key for a table.

5.6 Secondary Key

Secondary keys are candidate keys that are not selected to be the primary key.

5.7 Composite Key

A *composite key* is a combination of two or more fields in a table that can be used to uniquely identify each record in a table. Uniqueness is only guaranteed when the fields are combined. When taken individually, the fields do not guarantee uniqueness.

5.8 Foreign Key

A *foreign key* is an attribute (field) in one table that refers to the primary key in another table.

Explain the advantage of using a database rather than flat files. [2011]

Database provides referential **integrity**, every time a set of data is entered, it verifies if the data is a valid entry.
Database provides data **privacy**, it limits users to only access part of the data that is necessary and visible to them. Thus, only authorized
Database provides data **confidentiality**, it limits users to only access part of the data that is necessary and visible to them. Thus, only authorized users will have access to the relevant data.
Database minimize data **redundancy** by introducing normalization to the relations, and hence the data integrity will be enhanced.
Database provides data **independency**; such that program can continue to function even if the internal structure of the data storage changes.
Database also provides data **consistency**, because if one file inside one of the computer changes, and the other files in other computers are not updated, errors will arise. However, a centralized DBMS will ensure that the data is consistent across the views of different computers.

6. Computer Networks

6.1 Networking Devices

Describe two features of the router. [2017]

1. Forwards data packets between computer networks and perform the traffic directing functions on the Internet
2. Maintains a route table so that it can forward the data packets to the correct interface based on the destination address in the IP header. A router works on OSI layer 3.

Router vs. Switch vs. Hub

Hub	Switch	Router
OSI Layer 1 – Physical Layer	OSI Layer 2 – Data Link Layer	OSI Layer 3 – Network Layer
Connects different devices to form a network		Connects 2 or more different networks
Broadcasts to all connected devices.	Transmits data to only the devices intended to listen. Manages the MAC addresses of devices connected to the local area network.	Maintains a route table so that it can forward the data packets to the correct interface based on the destination address in the IP header.

How does a router act as a gateway to the Internet?

The Internet is made up of a massive network of specialized computers called *routers*. Each *router's* job is to know how to move *packets* along from their source to their destination. A *packet* will have moved through multiple *routers* during its journey. A *router* maintains a route table so that it can forward the data packets to the correct interface based on the destination address in the IP header.

6.2 Client-Server Model

A *client-server model* is a distributed application structure that **partitions tasks or workloads** between the **providers of a resource or service**, called servers, and **service requesters**, called clients. Clients do not share any resources but requests a server's content or service function.

A *client* is a hardware or software that **makes requests to the server** through a network.

A *server* is host machine (which can be a combination of both hardware and software) **which receives and processes requests** and **transmits data** to other computers in the network. Its main purpose is to store files, applications, services and hardware and allow their access to clients through the network, and back-end processing.

Examples: Email, network printing, World Wide Web.

6.3 Role-based Authorisation

Describe a method that can be used to ensure only office staff can change the system and only the principal can view confidential reports. [2017]

Authorization is the mechanism by which you control the operations and resources an authenticated client can access. Role-based authorization controls can be enforced for office staff and the principal such that only office staff can change the system and only the principal can view confidential reports after logging in with their user ID and password.

The Principle of Least Privilege states that a subject should be given only those privileges needed for it to complete its task.

6.4 Error Detection

Explain what is meant by even parity and why it is necessary. [2013]

- In *even parity*, one bit is used as a parity bit to ensure that the number of 1s in the data is even as a form of error detection mechanism. The parity bit will be 0 for data which has an even number of 1s, 1 for data which has an odd number of 1s.
- Even parity is necessary to detect whether the data sent is the intended, correct data, because there might be transmission errors occurring when sent through the network, or interception by external parties (hackers), causing the number of 1s to become odd if one of the bits is changed. Errors with 2 or more bits (in even numbers) changed cannot be detected

Describe two checks that the receiving computer should make for the integrity of 1) the individual bytes which make up a packet and 2) the collection of bytes which makes up the packet. [2016]

Describe three validation checks the receiving computer should perform on the packet. [2016]

- **Format check**: Ensure that every packet message begins and ends with a # character
- **Range check:** Ensure that every packet consists of only space characters, # characters, or uppercase letters. If other characters are detected, the packet is invalid**.**
- **Type check:** Ensure that the data type of each packet is in string format (not integer, float, etc)
- **Presence check:** Ensure that there are no empty fields, packet should not be empty

An email packet consists of 128 bytes. The first 126 bytes contain both control data and also a part of the email message. Byte 127 in the packet is a checksum. Byte 128 is currently not used. Even parity is used in each byte. [SP9597]

(a) A computer receives a packet which contains the following byte: 0 1 1 0 1 1 1 0. Describe the actions of the computer upon receipt of this byte.

Check that parity of byte is correct i.e. there should be an even number of '1's .
If there is an odd number of 1s, there is a problem with parity. The computer issues a request to server to retransmit byte / reports error to server.

(b) Describe how the checksum byte is produced by the sender computer.

The first 126 bytes are put through an algorithm that adds the contents together (possibly with positional weightings). Any overflow is discarded so that result is one byte value.

Describe how a checksum can be used to detect errors in data transmission. [2009]

- Sender computes checksum of data using some encrypted aggregate function on data e.g. MD5 or SHA1
- Sender transmit data and checksum separately to receiver
- Receiver runs same checksum generator on data and verifies that both transmitted and generated checksums match each other
- If a data file is corrupted, small variances produces large differences in the checksum
- e.g. software programs distributed via the Internet often comes with a check sum for the receiver to verify the file's integrity

Echo Check

With an echo check, the receiving computer sends a copy of the data immediately back to the sending computer for comparison.
The sending computer compares the two sets of data to check if any errors occurred during the transmission process.
If an error has occurred, the data will be transmitted again.

6.5 Firewall

Describe two ways that a firewall can be used to block unauthorised access to a network. [2018]

1. *Packet filtering* is a firewall technique used to control network access by monitoring outgoing and incoming packets and allowing them to pass or halt based on the source and destination Internet Protocol (IP) addresses, protocols and ports. If the information packet doesn't pass the inspection, it is dropped.
2. *Proxy firewalls / application-level gateways* operate at the application layer. These firewalls are delivered via a cloud-based solution or another proxy device. Rather than letting traffic connect directly, the proxy firewall first establishes a connection to the source of the traffic and inspects the incoming data packet. Once the check is complete, and the packet is approved to connect to the destination, the

proxy sends it off. This creates an extra layer of separation between the "client" (the system where the packet originated) and the individual devices on the network.

Describe two methods that could be used to restrict access to inappropriate websites. [2018]

- Keep track of a list of common inappropriate websites and block all such web request using a firewall that sits between LAN and WAN
- Install a firewall that sits between LAN and Wan to search for inappropriate words in both web request and web content retrieved in html document from unknown web server (e.g. gambling, sex, ToTo etc)
- Block all VPN connections

6.6 Data Loss

Data loss is an error condition in information systems in which information is destroyed by failures or neglect in storage, transmission, or processing. Information systems implement backup and disaster recovery equipment and processes to prevent data loss or restore lost data.

Note: Data loss is also distinct from data breach, an incident where data falls into the wrong hands (but still not lost), although the term data loss has been used in those incidents.

- Schedule back up regularly to offsite data store. E.g. VMWare
- Do cloud backup. E.g. CloudBerry
- Using 3-2-1 strategy, keep 3 copies of data, 2 local and 1 offsite.

6.7 Synchronous & Asynchronous Data Transmission

Synchronous transfer is the data transfer method that sends a continuous stream of data to the receiver using regular timing signals that ensures both transmitter and receiver are synchronized with each other.

Conversely, *Asynchronous* Data Transfer is the data transfer method that sends data from transmitter to receiver with parity bits (start and stop bits) in uneven intervals.

For example, chat rooms and video conferencing use synchronous data transfer while emails use asynchronous data transferring.

Which of the following diagrams indicate asynchronous data transmission? [2017]

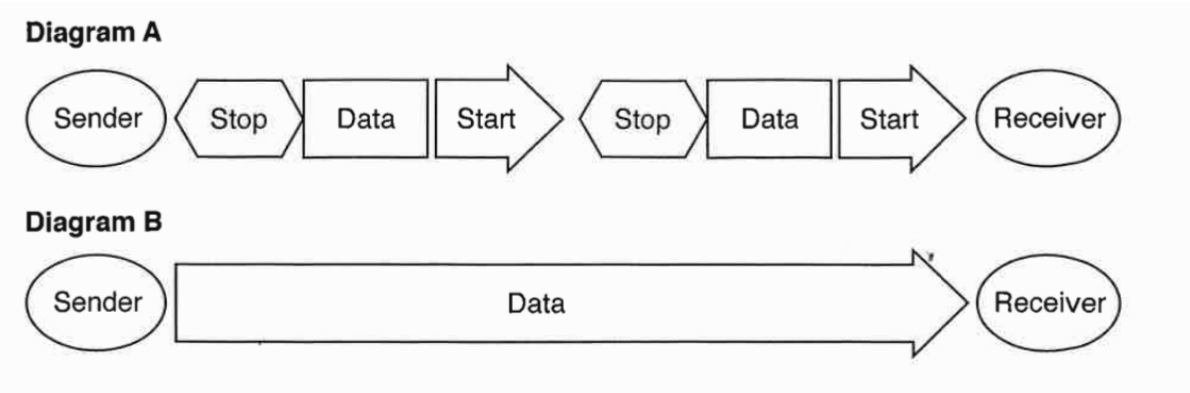

Diagram A. In asynchronous communication, a sender inserts special start and stop bit patterns between each byte of data. This enable the receivers to distinguish the bytes in the data stream.

Explain why asynchronous data transmission affects network performance. [2017]

In asynchronous communications, large relative overhead is needed, and a high proportion of the transmitted bits are uniquely for control purposes and thus carry no useful information. This therefore affects network performance.

Explain the differences between synchronous and asynchronous data transmission.

1. In synchronous transmission data is transmitted in the form of chunks, while in asynchronous transmission data is transmitted one byte at a time.
2. Synchronous transmission needs a clock signal between the source and target to let the target know of the new byte. In comparison, with asynchronous transmission, a clock signal is not needed because of the parity bits that are attached to the data being transmitted, which serves as a start indicator of the new byte.

3. The data transfer rate of synchronous transmission is faster since it transmits in chunks of data, compared to asynchronous transmission which transmits one byte at a time.
4. Asynchronous transmission is straightforward and cost-effective, while synchronous transmission is complicated and relatively pricey.
5. Synchronous transmission is systematic and necessitates lower overhead figures compared to asynchronous transmission.

6.8 Server-Side Scripting

Server-side scripting is a technique used in web development which involves employing scripts on a web server which produce a response customized for each user's request to the website. The alternative is for the web server itself to deliver a static web page.

Explain how the web server uses server side script to process the form. [2018]

- A POST request will be received by the server-side script.
- Data is read from the POST request form submission.
  Data verification to be performed against the current database to see if the user already exists.
- Data to be processed and updated into the database system at the server side.
- After success or unsuccessful registration, a GET request to display the feedback of the user will be generated and pushed to the client browser.

HTML request methods:

- The GET method requests a representation of the specified resource
- The POST method requests that the server accept the entity enclosed in the request as a new subordinate of the web resource identified by the Uniform Resource Identifier (URI). The data POSTed might be, for example, an annotation for existing resources; a message for a bulletin board, newsgroup, mailing list, or comment thread; a block of data that is the result of submitting a web form to a data-handling process; or an item to add to a database.
- The PUT method requests that the enclosed entity be stored under the supplied URI. If the URI refers to an already existing resource, it is modified; if the URI does not point to an existing resource, then the server can create the resource with that URI

6.9 Cloud Computing

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand.

Advantages

- Avoid up-front infrastructure costs e.g. purchasing servers
- Allows organizations to focus on their core businesses instead of spending time and money on computer infrastructure
- Allows enterprises to get their applications up and running faster
- IT teams could more rapidly adjust resources to meet fluctuating and unpredictable business demand

Disadvantages

- Limited customization options (no control of the back-end infrastructure)
- Limits in the form of data caps on the maximum bandwidth allocated to a customer
- 'Pay as you go' model may lead to unexpectedly high charges if administrators do not adapt to cloud pricing model

Describe three economic benefits to the organisers of using cloud-based storage. [2018]

- Cloud storage allows organisers to bypass large up-front infrastructure costs e.g. purchasing hard drives. Even if organisers are able to purchase such hardware, cloud storage has a low maintenance cost compared to organisers having to maintain the servers themselves.
- Cloud storage is reliable and provides strong data security, hence it saves cost of investing in data security features compared to private data storage options.
- Cloud storage have flexible pricing models that charge a monthly fee, which are comparatively cheaper especially if the organisers do not need the data anymore after this one-time event. Furthermore, they charge a fee based on the amount of storage actually used, so it may be more cost-efficient than investing in hardware but ending up with a lot of excess capacity.
- Cloud access to important files gives users the freedom and flexibility to work from anywhere. Leveraging cloud storage solutions as part of a mobile workforce improves productivity by allowing users to work more efficiently with their files. Whether it is on their laptop or smartphone, cloud-based storage solutions improves overall accessibility in the workplace.

7. Project Management

7.1 Purpose of Project Proposal

Ensures that the project is **well-defined**, sets the **requirements** and **checkpoints** of the project, and provides **identification and management of possible risks** of the project. It describes the execution, management and control of the project:

- Who: Allocation of job scopes & organisation of team members
- What: Scope of the project & work to be done to achieve the desired project outcome
- Why: Rationale of the project
- When: Project timeline to set deadline for tasks
- How: Process of doing the project

It also acts as a **single point of reference** for everyone in the team.

Note: The project proposal may be done by the software house to be submitted to client for approval, or done by client to be submitted to software houses who will express their interest. Depending on the context, the purpose differs.

State three members of the project team. Describe the role of each member. [2018]

Project Manager	• Prepare project plans, schedule and budget • Works with business analyst to prepare user specification document • Works with system analyst in design • Coordinate resources when needed • Monitors issues and changes requests • Prepares project status report
System Analyst	• Gathers requirements from users and converts them into specifications so that the test plan can be designed • Design all technical aspects of the system based on user specifications • Assist business analyst to build logical data models • Prepare technical documentation • Evaluate and modify code at times • Coordinate system testing • Monitor technical issues arising during implementation
QA Manager and Testers	• Conduct functional/integration testing • Conduct UAT • Ensure software can run on different platforms based on requirements e.g. if it is a web application, it should run on different browsers
UI/UX Designers (front-end developers)	• Conduct UI/UX research • Design and develop the actual client-side web application • Refine and improve design of web interfaces
Database Developer	• Integrate database with back-end server • Communicate with back-end developers to ensure that they know how to use the database
Back-end Developer	• Develop the server-side of the web application • Ensure that data is handled correctly and securely

7.2 Specifications

Note: These are also the key deliverables of the Analysis phase.

Research Report

Existing System Review
Allows users to verify the accuracy of the model created.

- Information on existing system (what and how it is being done)
- Diagrams to understand the system e.g. DFD
- Set of problem statements of existing software (to determine new system requirement)

New System Requirement
This should consider the technical, economic, legal and social factors.
It must meet the URS requirements.

- Set of system objectives (what the new system must be able to do)
- Modify current system to satisfy problem statements

- Recommended course of action

Feasibility Report

Shows whether the project is likely to succeed within its constraints. It contains cost-benefit evaluation and recommendation of whether the system is worth developing.

- **Product:** A general statement of the product; give a brief description of what the proposed system will do, highlighting where the proposed system meets the specified business requirements of the organisation.
- **Technical Feasibility:** Will the proposed system perform to the required specification? Outline technical systems options you propose to use, which will give a technical solution satisfying the requirements and constraints of the system, as outlined in the terms of reference.
- **Social Feasibility:** Consideration of whether the proposed system would prove acceptable to the people who would be affected by its introduction. Describe the effect on users from the introduction of the new system; consider whether there will be a need for retraining the workforce. Will there be a need for relocation of some of the workforce? Will some jobs become deskilled? Will the current workforce be able to perform effectively any new tasks introduced by the proposed system? Describe how you propose to ensure user co-operation before changes are introduced.
- **Economic Feasibility:** Consider the cost/benefits of the proposed system. Detail the costs that will be incurred by the organisation adopting the new system; consider development costs and running costs. Detail benefits that the new system will bring, direct economic benefits such as reduced costs, and indirect benefits, such as improved management information and better customer service. Illustrate the cost/benefit of the new system by applying a suitable cost/benefit analysis method such as the payback method.
- **Market Research:** A comprehensive market research identifying a need for the product. Detail all market research you carried out, listing sources of information. Justify any conclusions you have drawn from your research. Identify the potential customer base for your product, together with evidence of customer need for the product. Describe how you propose to compete with similar products on the market.
- **Alternative Solution:** Consideration of alternative solutions should be documented. At least two alternative business or technical systems options should be considered. Detail the differences between these options and the proposed system. Justify your choice of the proposed system and the reasons for rejecting the alternative options.

User Requirements Specification (URS)

A detailed description of the user requirements of the new system after research and analysis.

Program Requirements – the functions and workflow that the system must be able to perform
Data Requirements – the type of information that a system must be able to process
Life Cycle Requirements – including how the system will be maintained and users trained

Software Requirements Specification (SRS)

- It is the description of a software system that is to be developed.
- It lays out **functional** and **non-functional** requirements.
- It establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function.
- It is a rigorous assessment of requirements before the more specific system design stages.
- Its goal is to **reduce** later **redesign**.

The documents are submitted with the research report for approval from key stakeholders to allow the project to proceed.

The initial phase of the system development cycle required that a specification be created for the *system*. Describe two sections of the report. [2018]

Functional Requirement

- Major functions provided by the software
- *Example:* when users check credit balance, a graph and the credit balance for past 3 months must be displayed by default. Options for all time past year and past 6 months are also available.

Non-Functional Requirement

- Quality of functions provided by the software
- *Example:* The system can update the displayed credit balance record correctly every 5 minutes
- *Example:* During login process, the software must prompt user for 2FA. Users must login with their own ID and password, followed by OTP sent to the mobile phone

Hardware Requirement

- Minimum hardware/software specifications to run the application

7.3 Modular Program Development

Main functions of the system are **designed separately** as modules, such that they **do not require one and another to function** and are able to function perfectly on their own.

What are some benefits of modular design?

- As the main functions of the system are designed separately as modules, they do not require one and another to function and are able to function perfectly on their own. Every module can progress, and it will not delay other modules as they are independent. This helps project manager to efficiently allocate human resources as no module has to wait for other modules to complete.
- Modules can be reused by other similar projects
- Minimize bugs in the future during integration testing

The PERT chart indicates that some testing can commence almost as soon as the program development does. Describe the type of program development that would allow for this. [2016]

Main functions of the system are designed separately as modules, such that they do not require one and another to function and are able to function perfectly on their own.
Therefore, as every module of the system is developed separately and can function independently between one another, initial testing can commence on the first complete module, while the other modules of the program are still in development.

7.4 Test-Driven Development

Test-driven development (TDD) relies on the repetition of a very short development cycle: software is tested against very specific test cases, and improved to pass new test cases. (opposed to software development that allows software that has not yet been proven to meet previous requirements to be added)

7.5 Top-Down Design

Top-down design involves first an overview of the system, specifying but not detailing any first-level subsystems. Each subsystem is then refined in greater detail until the entire system is reduced to its base elements.

7.6 Bottom-Up Design

Bottom-up design involves first the individual base elements of the system specified in great detail. They are then linked together to form the entire system, sometimes in many levels.

7.7 Design Techniques

1. Top-down design
2. Bottom-up design
3. Modular design
4. Test-driven development (TDD)

Explain why it is important for each member in the design team to use the same technique. [2018]

- This allows different modules, sub systems to be developed concurrently.
- It also ensures that testing can be done at the same pace e.g. bottom-up testing, higher-level components only tested after lower-level units are developed. If different techniques are used, testing cannot be done in a coordinated manner.
- If not, there will be disruption during the development phase as integration between sub systems may have issues.
- Progress of development cannot be measured if members use different techniques e.g. clear progression from unit testing → integration testing → functional testing → system testing → UAT.

7.8 Design Checks

Describe the checks that the team needs to make at the end of the design stage. [2017]

- **System design check:** Check the deliverables of the software design stage e.g. ER diagram, database schema, UML class diagram, interface design, algorithm design. These designs need to be checked for correctness and/or efficiency, usability, etc. either manually by domain specific experts or by members using semi-automated or automated tools.
- **Hardware check:** Check whether hardware e.g. hard disk, computer and card reader etc to be purchased is compatible with the software used, and is able to handle the required number of users.
- **Budget check:** Check whether budget available is sufficient to implement the system designed.

- **Human resource check:** Check whether it is feasible to implement the design given the human resource available.

Describe two methods that could be used to check this design. For each method, identify the members of the development team involved other than the system analyst.

- ER diagram and database schema – *database administrator* will need to check that the normalised design adheres to third normal form (1NF, 2NF and no non-key/transitive dependencies), no many-to-many relationships in the ER diagram, table specification contains appropriate field names and data types, with primary key, valid integrity (e.g. non-null fields, default values, etc.) and referential constraints (foreign key), as well as multi-level security access to the tables and reports
- Interface design – *user interface (UI) designer* will need to check that input forms contain appropriate controls, navigation and labels are clear and consistent, useful prompts for intrusion/error messages and confirmation actions, appropriate use of colours and contrast with corporate style and identity, minimise number of steps/clicks to perform actions, input data validation, etc.
- UML class diagram – object-oriented analysis and design trained *developer* to check for well-designed classes with private data and public methods to ensure encapsulation, inheritance to promote code reuse and facilitate modification/maintenance, polymorphic method names across related classes for code generalisation, meaningful identifier names and community adopted case and naming conventions, etc.

7.9 PERT Chart

Explain the significance of the dashed lines on the PERT chart. [2016]

Dashed lines on a PERT Chart indicate a dummy activity. Dummy activities are imaginary activities without any time duration to demonstrate relationships/dependencies between activities that would otherwise be difficult to show with simple arrow linkages.

7.10 Documentation

Describe three sections that should be included in the user guide for this system. [2017]

Purpose
- Description of what the system is designed to do

Minimum hardware and software requirements of the system

Installation procedure
- Instructions on how to load and run the system

Detailed instructions on how to operate main/important/every part of the system
- with screenshots as visual aid, showing the system in typical use
- example inputs and outputs, within the screenshots

Troubleshooting Guide
- explanations of error messages, their meaning and how to deal with them

Frequently Asked Questions (FAQ)