Computer Systems Question Bank

Computer Science Class XI (1st Year), Sindh Textbook Board, Unit 1. Short questions, differences, long questions and practical questions with answers, built from the new STBB textbook and its end-of-unit exercise.

STBB Computer Science XI | Unit 1, pages 1 to 70 | Digital logic, SDLC, OSI and TCP/IP | Moderate level

Read this first. This unit has four big areas: digital data and Boolean logic (gates, truth tables, K-maps), software development (SDLC, Waterfall, Agile), and communication models (OSI, TCP/IP). Learn the seven gate truth tables, the six SDLC phases, the seven OSI layers with the mnemonic, and the difference tables, because these are the most repeated in board papers. Tags: **BOOK EXERCISE** = taken from the end-of-unit exercise, **LIKELY** = from the book topics.

Short questions

3 to 4 lines each

S1 Define a digital system. Why does it use only 0 and 1?

BOOK TOPIC 1.1.2 | LIKELY

A digital system is an electronic system that works with only two values, 0 and 1, which are called binary digits or bits. Here 0 represents OFF, LOW or FALSE and 1 represents ON, HIGH or TRUE. Only two states are used because it is very easy for an electronic circuit to tell the difference between two levels, so the system becomes simple, accurate and not easily affected by noise.

S2 Define digital logic and give two examples of digital devices.

BOOK EXERCISE 3(I) | LIKELY

Digital logic is the use of binary values (0 and 1) and logic operations such as AND, OR and NOT to process, store and control information inside electronic circuits. All modern electronic devices are built on it. Examples of digital devices are a computer, which processes data as 0s and 1s, and a smartphone, which stores photos and music as binary data. A digital watch, a calculator and a digital camera are also digital devices.

S3 Write any four advantages of using digital systems.

BOOK TOPIC 1.1.2 | LIKELY

- Reliable: digital signals are less affected by noise and interference, so the data stays correct during transmission.
- Accurate: there is no loss of quality when data is copied again and again.
- Easy to design: only two states (0 and 1) have to be handled, so circuits are simple to build.
- Programmable: digital systems can be controlled easily by software, so one device can do many tasks.

S4 What is a signal? Name the two main types of signals.

BOOK TOPIC 1.1.3 | LIKELY

A signal is a function of one or more variables that indicates a physical phenomenon, and it works as a carrier of information. Signals can convey different types of information depending on the application, such as sound, video or data. The two main types of signals are analog signals, which are continuous and take any value in a range, and digital signals, which have only two levels, HIGH (1) and LOW (0).

S5 What is Boolean algebra? Who developed it?

BOOK TOPIC 1.1.4 LIKELY

Boolean algebra is a branch of mathematics that deals with only two values, TRUE (1) and FALSE (0). It was developed by George Boole, an English mathematician, in the 1850s. It is the foundation of all digital electronics and computer programming, and it uses variables, constants, operations (AND, OR, NOT) and expressions to show how logic operations are applied.

S6 State the AND operation with its symbol and expression.

BOOK TOPIC 1.1.5 LIKELY

The AND operation gives output 1 only when all the inputs are 1; if any input is 0 the output is 0. Its symbol is a dot (.) or the wedge sign, and its Boolean expression is Y = A . B (read as A AND B). A real-life example is a door with two locks: the door opens only when both locks are unlocked.

S7 State the OR operation with its symbol and expression.

BOOK TOPIC 1.1.5 LIKELY

The OR operation gives output 1 when any one input is 1, and the output is 0 only when all the inputs are 0. Its symbol is a plus sign (+), and its Boolean expression is Y = A + B (read as A OR B). A real-life example is a room with two doors: we can enter the room if either door is open.

S8 What is the NOT operation? Why is it called a unary operation?

BOOK TOPIC 1.1.5 LIKELY

The NOT operation inverts (complements) the input: if the input is 1 the output is 0, and if the input is 0 the output is 1. It is written as Y = A' (or the bar sign over A). It is called a unary operation because it works on only one operand, unlike AND and OR which need two or more inputs. A light switch is a real-life example.

S9 What is a truth table? Write the steps to create it.

BOOK TOPIC 1.1.6 LIKELY

A truth table shows all the possible input combinations of a logic gate or circuit and the corresponding output for each combination. To create it: first count the number of inputs (n), then calculate the total rows as 2 to the power n, then list all the input combinations in binary order, and finally calculate the output for each combination. For example, 3 inputs give 8 rows.

S10 Why are NAND and NOR called universal gates?

BOOK TOPIC 1.1.7 LIKELY

A universal gate is a gate that can be used alone to build every other gate. NAND and NOR are called universal gates because the basic gates AND, OR and NOT can all be made using only NAND gates, or using only NOR gates. This is useful in practice because a circuit can be built from one type of gate, which reduces cost and makes manufacturing simpler.

S11 **What is the operator precedence order in a logic expression?**

BOOK TOPIC 1.1.8 LIKELY

When a logic expression has more than one operator, the operations are solved in a fixed order of priority. The first priority is given to parentheses, the second to NOT, the third to AND and the last to OR. For example, in Y = A' + B . C the NOT of A is solved first, then B . C, and at the end the OR is applied.

S12 **Define minterm and maxterm.**

BOOK TOPIC 1.1.10 LIKELY

A minterm is a product (AND) of all the variables in which each variable is either complemented or not, and it gives output 1 for exactly one combination of inputs. A maxterm is a sum (OR) of all the variables and it gives output 0 for exactly one combination of inputs. Minterms are used in the SOP (Sum of Products) form, for example A'B + AB.

S13 **What is a logic diagram? Write the steps to draw it.**

BOOK TOPIC 1.1.11 LIKELY

A logic diagram is a graphical representation of a logic expression in which logic gates are connected to create a circuit. Steps: identify the expression, break it into parts according to the precedence of operations, draw the gate for each part (highest precedence first), and connect the gates with lines. For example, Y = A . B + C uses an AND gate whose output goes into an OR gate with C.

S14 **What is a Karnaugh map (K-map)? Why is it used?**

BOOK TOPIC 1.1.12 LIKELY

A Karnaugh map (K-map) is a grid-based graphical tool that is used to simplify Boolean expressions. It is used because algebraic simplification can be long and difficult, while a K-map gives a visual and systematic way to find the simplest expression. A simpler expression needs fewer gates, so the circuit becomes cheaper, smaller and faster.

S15 **Why is the K-map column order 00, 01, 11, 10 and not 00, 01, 10, 11?**

BOOK TOPIC 1.1.12 LIKELY

The columns of a K-map follow the Gray code, in which only one bit changes between any two neighbouring cells. This means that adjacent cells differ by only one variable, and only such cells can be grouped together so that the changing variable can be cancelled. If the normal binary order were used, neighbouring cells could differ in two bits and the grouping would give wrong results.

S16 **What is LogiSim? Name any two things you can do with it.**

BOOK TOPIC 1.1.13 LIKELY

LogiSim (LogiSim Evolution) is a free graphical tool used for designing and simulating logic circuits. We can place logic gates, input pins and output pins on the drawing canvas and connect them with wires to build a circuit. We can then use the Poke Tool to change the input values and observe the output, which helps us verify the truth table of the circuit.

S17 **What is SDLC? Write its full form.**

BOOK TOPIC 1.2.2 LIKELY

SDLC stands for Software Development Life Cycle. It is a structured process that guides software development from the beginning to the end and gives a clear framework with specific phases and deliverables. It makes sure that the software is planned, designed, built, tested, released and maintained in an organized way, so that the final product meets the needs of the users.

S18 **Write any four benefits of using SDLC.**

BOOK TABLE 1.12 LIKELY

- Clear goals: everyone in the team knows what is to be built.
- Better planning: time and resources are managed well.
- Quality assurance and risk reduction: testing means fewer bugs and problems are found early.
- Documentation and team communication: records help in future maintenance and everyone follows the same process.

S19 **Name the six phases of SDLC in order.**

BOOK TOPIC 1.2.3 LIKELY

The six phases of SDLC are: 1. Requirement Analysis (Planning), where the team gathers what the software should do; 2. Design, where the structure and screens are planned; 3. Implementation (Coding), where the code is written; 4. Testing, where bugs are found and fixed; 5. Deployment, where the software is released to users; and 6. Maintenance, where support and updates continue.

S20 **What is SRS? Which SDLC phase produces it?**

BOOK EXERCISE 2(IV) LIKELY

SRS stands for Software Requirements Specification. It is a document that lists all the requirements and features of the software, for example that students should see their grades and teachers should enter attendance. It is the deliverable of the first phase, Requirement Analysis (Planning), and it is used by the team in all later phases as a reference.

S21 **What is black box testing? Give an example.**

BOOK TOPIC 1.2.3 LIKELY

In black box testing the tester does not see or know the code of the software. The tester only gives inputs and checks the outputs to find out whether the software works correctly for the users. It tests what the software does and not how it does it. Example: entering a username and password and checking that the login works, and that a wrong password shows an error.

S22 What are the four types of software maintenance?

BOOK EXERCISE 4(V) LIKELY

- Corrective: fixing bugs that are found after release, for example fixing a login error.
- Adaptive: changing the software for a new environment, for example updating it for a new Windows version.
- Perfective: adding new features or improving the software, for example adding a dark mode.
- Preventive: preventing future problems, for example improving the code for better performance.

S23 What is the Waterfall model? When is it used?

BOOK TOPIC 1.2.4 LIKELY

The Waterfall model is a linear and sequential approach in which each phase must be completed before the next phase begins, and there is no going back to an earlier phase. It is used when the requirements are clear and fixed, the project is small to medium in size, the technology is well understood and no major changes are expected during development.

S24 What is the Agile model? What is a sprint?

BOOK TOPIC 1.2.4 LIKELY

The Agile model is an iterative and incremental approach in which software is developed in small cycles instead of one big plan. Each short cycle, usually 2 to 4 weeks long, is called a sprint. At the end of every sprint a working part of the software is delivered, the customer gives feedback, and changes are included in the next sprint. Scrum is a popular Agile framework.

S25 What is a communication model? Why do we need one?

BOOK TOPIC 1.3.1 LIKELY

A communication model is a set of rules and layers that explains how data travels from one device to another over a network, like the rules of a postal system. We need it so that computers of different companies and countries can talk to each other, problems in a network are easier to find, and a complex task is divided into smaller steps called layers. The two main models are OSI and TCP/IP.

S26 Write the full forms of OSI and TCP/IP.

BOOK TOPIC 1.3.1 LIKELY

OSI stands for Open Systems Interconnection. It is a 7-layer reference model developed by ISO in 1984 and is mainly used for learning and design. TCP/IP stands for Transmission Control Protocol / Internet Protocol. It is a 4-layer model developed in the 1970s by the US Department of Defense, and it is the model that the Internet actually uses today.

S27 Write the seven OSI layers from bottom to top with the memory mnemonic.

BOOK TABLE 1.31 LIKELY

From bottom to top the layers are: 1. Physical, 2. Data Link, 3. Network, 4. Transport, 5. Session, 6. Presentation and 7. Application. The mnemonic to remember them is "Please Do Not Throw Sausage Pizza Away", where the first letter of each word is the first letter of a layer (P, D, N, T, S, P, A). The bottom layers deal with hardware and the top layers with software.

S28 **What is the job of the Application layer? Name four protocols used at this layer.**

BOOK TABLE 1.32 LIKELY

The Application layer is the topmost layer and the closest to the end user. It provides network services directly to the user's applications, such as a web browser or an email client, and it identifies the communication partners. The protocols used at this layer are HTTP (web pages), FTP (file transfer), SMTP (sending email) and DNS (converting domain names into IP addresses).

S29 **What is the job of the Transport layer? Name its two protocols.**

BOOK TOPIC 1.3.3 LIKELY

The Transport layer is responsible for reliable end-to-end communication between two devices. It breaks large messages into smaller segments, makes sure that they all arrive correctly, controls the flow of data and reassembles the segments at the destination. It also uses port numbers to identify the application. Its two protocols are TCP (reliable) and UDP (fast).

S30 **What is the job of the Network layer? Name two of its protocols.**

BOOK TOPIC 1.3.3 LIKELY

The Network layer is responsible for logical addressing and routing. It assigns IP addresses to devices and decides the best route for the data to travel from the source to the destination, even across many different networks. Routers work at this layer. Two of its protocols are IP (IPv4 and IPv6) and ICMP, which reports errors and is used by the ping command.

S31 **What is the difference between an IP address and a MAC address?**

BOOK EXERCISE 3(IV) LIKELY

An IP address is a logical address used to identify a device on a network and to route data between networks. It can change, for example 192.168.1.1 (IPv4), and it works at the Network layer. A MAC address is a unique physical hardware address permanently assigned to the network interface card, for example AA:BB:CC:DD:EE:FF, and it works at the Data Link layer inside the local network.

S32 **Define encapsulation and decapsulation.**

BOOK TOPIC 1.3.4 LIKELY

Encapsulation is the process of adding extra information, called a header, to the data at each layer as the data moves down the layers at the sender's side. Decapsulation is the reverse process: at the receiver's side each layer reads and removes its own header as the data moves up the layers, until the original data reaches the Application layer and is presented to the user.

Differences

3 to 5 marks each

D1 Differentiate between analog and digital signals.

BOOK EXERCISE 3(II) LIKELY

Point	Analog signal	Digital signal
Nature of data	Continuous, takes any value	Discrete, only two levels
Unit	Physical quantities like voltage	Binary digits (bits) 0 and 1
Wave shape	Smooth sine wave	Square wave
Range of values	Infinite values	Only two levels (HIGH and LOW)
Noise	Easily affected by noise	Very resistant to noise
Examples	Human voice, thermometer	Computer data, smartphone, digital watch

D2 Differentiate between AND, OR and NOT operations.

BOOK TOPIC 1.1.5 LIKELY

Point	AND	OR	NOT
Symbol	Dot (.)	Plus (+)	Bar or prime (')
Expression	Y = A . B	Y = A + B	Y = A'
Inputs	Two or more	Two or more	Only one
Output is 1 when	All inputs are 1	Any input is 1	Input is 0
Real-life example	Door with two locks	Room with two doors	Light switch

D3 Differentiate between minterm and maxterm.

BOOK TOPIC 1.1.10 LIKELY

Point	Minterm	Maxterm
Operation	Product (AND) of variables	Sum (OR) of variables
Gives	Output 1 for one combination	Output 0 for one combination
Used in	SOP form	POS form
Example (A = 0, B = 1)	A'B	A + B'

D4 Differentiate between black box and white box testing.

BOOK TABLE 1.14 LIKELY

Black box testing	White box testing
Does not need code knowledge	Needs programming knowledge
Tests functionality	Tests code structure
Done by testers	Done by developers
Tests what the software does	Tests how the software does it

D5 Differentiate between Waterfall and Agile models.

BOOK EXERCISE 3(III) LIKELY

Point	Waterfall	Agile
Approach	Linear and sequential	Iterative and incremental
Requirements	Fully defined before coding	Partially defined, evolve over time
Client involvement	At the start and the end	Continuous, after every sprint
First working product	After the full project (about 6 months)	After the first sprint (about 2 weeks)
Response to change	Difficult and costly	Easy, done in the next sprint
Documentation	Heavy	Light, working software preferred
Testing	One phase after coding	Continuous in every sprint
Suitable when	Requirements are fixed	Requirements are unclear or change

D6 Differentiate between TCP and UDP.

BOOK EXERCISE 3(V) LIKELY

Point	TCP	UDP
Full name	Transmission Control Protocol	User Datagram Protocol
Type	Connection-oriented, reliable	Connectionless, fast
Delivery	Guarantees delivery, retransmits lost data	No guarantee of delivery
Speed	Slower, higher overhead	Faster, lower overhead
Used for	Web browsing, email, file transfer	Video streaming, online gaming, live calls

D7 Differentiate between encapsulation and decapsulation.

BOOK TOPIC 1.3.4 LIKELY

Encapsulation	Decapsulation
Adding headers to data	Removing headers from data
Happens at the sender	Happens at the receiver
Data moves down from layer 7 to layer 1	Data moves up from layer 1 to layer 7

D8 Differentiate between the OSI model and the TCP/IP model.

BOOK TABLE 1.37 LIKELY

Aspect	OSI model	TCP/IP model
Number of layers	7	4
Developed by	ISO (1984)	US Department of Defense (1970s)
Primary use	Reference model for learning and design	Practical model used on the Internet
Application layer	Three separate layers (Application, Presentation, Session)	Combined in one Application layer
Lower layers	Data Link and Physical are separate	Combined into Network Access layer
Protocol independence	Model defined first, protocols fit it	Built around existing protocols
Still used today	For teaching and troubleshooting	Yes, powers the Internet

D9 Differentiate between a hub, a switch and a router.

LIKELY CONCEPT

Device	OSI layer	Function
Hub	Layer 1 (Physical)	Broadcasts data to all connected devices
Switch	Layer 2 (Data Link)	Sends frames to the correct device using MAC addresses
Router	Layer 3 (Network)	Forwards packets between different networks using IP addresses

Long questions

5 to 8 marks each

L1 Explain the seven logic gates with their symbols, Boolean expressions and truth tables.

BOOK EXERCISE 4(I) LIKELY

Logic gates are electronic circuits that perform Boolean operations on one or more binary inputs and produce a single binary output. They are the basic building blocks of all digital systems such as computers, calculators and mobile phones. Every gate has one or more inputs, only one output and a specific function. There are seven gates: AND, OR, NOT, NAND, NOR, XOR and XNOR.

1. AND GATE

The AND gate gives output 1 only when all the inputs are 1. Its expression is Y = A . B. It works like a door with two locks, which opens only when both locks are unlocked. Its symbol is a D-shaped gate.

2. OR GATE

The OR gate gives output 1 when any one input is 1, and 0 only when all inputs are 0. Its expression is Y = A + B. It works like a room with two doors, where we can enter if either door is open. Its symbol is an arrow-shaped (curved) gate.

3. NOT GATE (INVERTER)

The NOT gate has only one input and it inverts the input, so 0 becomes 1 and 1 becomes 0. Its expression is Y = A'. It works like a light switch. Its symbol is a triangle with a small circle (bubble) at the output.

4. NAND GATE

The NAND gate is the opposite of the AND gate (NOT AND). It gives output 0 only when all inputs are 1. Its expression is Y = (A . B)'. Its symbol is the AND gate with a bubble. It is a universal gate.

5. NOR GATE

The NOR gate is the opposite of the OR gate (NOT OR). It gives output 1 only when all inputs are 0. Its expression is Y = (A + B)'. Its symbol is the OR gate with a bubble. It is also a universal gate.

6. XOR GATE (EXCLUSIVE OR)

The XOR gate gives output 1 when its inputs are different. Its expression is Y = A XOR B. Its symbol is the OR gate with an extra curved line at the input.

7. XNOR GATE (EXCLUSIVE NOR)

The XNOR gate gives output 1 when its inputs are the same. It is the opposite of XOR, and its expression is Y = (A XOR B)'. Its symbol is the XOR gate with a bubble.

SUMMARY AND TRUTH TABLES (TWO INPUTS)

A	B	AND	OR	NAND	NOR	XOR	XNOR
0	0	0	0	1	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	1	0	0	0	1

For the NOT gate: input 0 gives output 1, and input 1 gives output 0.

Draw. The standard symbols: AND (D shape), OR (curved arrow shape), NOT (triangle with a bubble), NAND (AND with a bubble), NOR (OR with a bubble), XOR (OR with an extra curved line) and XNOR (XOR with a bubble).

NAND and NOR are called universal gates because any other gate can be built using only NAND gates or only NOR gates.

L2 Draw the truth table for Y = A' . B + C. Also write the steps used to create a truth table from an expression.

BOOK EXERCISE 4(II) LIKELY

A truth table lists all the possible combinations of the inputs of a logic expression and the output of the expression for each combination. It helps us to check what a circuit will do for every possible input before building it.

STEP 1: IDENTIFY THE VARIABLES

Look at the expression and count the different input variables. In Y = A'.B + C the variables are A, B and C, so there are 3 inputs.

STEP 2: CALCULATE THE NUMBER OF ROWS

The total number of rows is 2 to the power n, where n is the number of inputs. Here n = 3, so rows = 2 x 2 x 2 = 8.

STEP 3: LIST ALL INPUT COMBINATIONS

Write the inputs in binary counting order from 000 to 111 so that no combination is missed. This is done by making the last column alternate 0, 1, 0, 1, the middle column 0, 0, 1, 1, and the first column 0, 0, 0, 0, 1, 1, 1, 1.

STEP 4: CALCULATE THE OUTPUT

Solve the expression by operator priority: first the NOT (A'), then the AND (A'.B), and last the OR with C. Write each intermediate result in its own column so that mistakes are easy to find.

TRUTH TABLE

A	B	C	A'	A'.B	Y = A'.B + C
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	1	1	1

1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	0	0	0
1	1	1	0	0	1

Y is 1 when C = 1, or when A = 0 and B = 1.

L3 What is a K-map? Explain the steps to simplify a 2-variable expression with an example.

BOOK TOPIC 1.1.12 LIKELY

A Karnaugh map (K-map) is a grid-based tool used to simplify Boolean expressions without long algebraic work. It places the truth table values in a grid where neighbouring cells differ by only one variable, so that the groups of 1s can be seen easily. A simpler expression needs fewer gates, which makes the circuit cheaper and faster. A 2-variable K-map has 4 cells (2 by 2 grid).

STEP 1: DRAW THE K-MAP AND FILL THE VALUES

Draw the 2 by 2 grid with A on the rows and B on the columns, and write the output of the truth table in each cell (cell 0 is A'B', cell 1 is A'B, cell 2 is AB' and cell 3 is AB).

STEP 2: GROUP THE 1S

Make groups of adjacent 1s in sizes of 1, 2, 4 or 8 (powers of 2). Groups should be as large as possible, and a 1 can be used in more than one group.

STEP 3: CANCEL THE CHANGING VARIABLE

In each group, the variable that changes its value is cancelled, and the variable that stays the same is kept.

STEP 4: WRITE THE SIMPLIFIED EXPRESSION

Join the terms of all the groups with OR to get the final simplified expression.

EXAMPLE (OR GATE)

The truth table has Y = 1 for AB = 01, 10 and 11, so the expression is Y = A'B + AB' + AB.

A / B	B = 0	B = 1
A = 0	0	1
A = 1	1	1

- Group 1: the right column (B = 1). A changes and B stays 1, so the term is B.
- Group 2: the bottom row (A = 1). B changes and A stays 1, so the term is A.

```
Y = A + B
```

The simplified result matches the OR gate truth table, which verifies the answer.

L4 Simplify Y = A'B'C + A'BC + AB'C + ABC using a 3-variable K-map.

BOOK TOPIC 1.1.12 LIKELY

A 3-variable K-map has 8 cells (2 to the power 3) and is drawn as a 2 by 4 grid, with A on the rows and BC on the columns. The column order must be 00, 01, 11, 10 (Gray code), so that only one bit changes between neighbouring cells.

STEP 1: IDENTIFY THE MINTERMS

Write each term as a binary number: A'B'C = 001 (m1), A'BC = 011 (m3), AB'C = 101 (m5) and ABC = 111 (m7). These are the cells where the output is 1.

STEP 2: FILL THE K-MAP

Put a 1 in the cells of these minterms and 0 in the other cells. All four 1s are in the columns BC = 01 and BC = 11 (for both A = 0 and A = 1).

A / BC	00	01	11	10

A = 0	0	1	1	0
A = 1	0	1	1	0

STEP 3: GROUP THE 1S

The four 1s form one square group of 4 cells (two columns and two rows), which is the largest possible group.

STEP 4: WRITE THE SIMPLIFIED EXPRESSION

Inside this group A changes (0 and 1) and B changes (0 and 1), so both are cancelled. Only C stays 1 in all the cells, so it is kept.

```
Y = C
```

RESULT

The original expression had 4 terms and 12 literals, while the simplified expression has only 1 term and 1 literal. This shows how a K-map can reduce a big circuit to a single wire.

L5 Explain all phases of SDLC in detail with activities and deliverables.

BOOK EXERCISE 4(III) LIKELY

SDLC (Software Development Life Cycle) is a structured process that guides software development from the beginning to the end. It has six phases. Each phase has its own activities and produces a deliverable, which is the output handed over to the next phase.

PHASE 1: REQUIREMENT ANALYSIS (PLANNING)

This is the first and the most important phase, in which the team gathers information about what the software should do. Activities: meeting customers and users, understanding their problems and needs, listing the required features and writing the SRS (Software Requirements Specification) document. Example: for a school system, students should see their grades and parents should receive notifications. Deliverable: SRS document.

PHASE 2: DESIGN

Based on the requirements, the team decides how the software will work and look. Activities: creating High-Level Design (overall structure) and Low-Level Design (details of each component), designing the database and the user interface (screens). Tools used are flowcharts, UML diagrams, ER diagrams and wireframes. Deliverable: Design document.

PHASE 3: IMPLEMENTATION (CODING)

Developers write the actual code using the design documents. Activities: writing code in a programming language such as Python, Java or C++, following coding standards, using version control (Git) and documenting the code. Best practices are writing clean readable code, adding comments and reviewing code with team members. Deliverable: working code or software.

PHASE 4: TESTING

The software is tested to find and fix errors (bugs). Activities: unit testing by developers, integration and system testing by the testing team, and user acceptance testing by the end users. Black box and white box methods are used. Deliverable: test reports and bug-free software.

PHASE 5: DEPLOYMENT

The tested software is released to the users. It can be released by direct deployment (replace the old system at once), phased rollout (to groups gradually), pilot release (small group first) or parallel running (old and new systems together). Deliverable: live software available to the users.

PHASE 6: MAINTENANCE

After release the software needs ongoing support and updates. Activities: fixing bugs (corrective), changing for new environments (adaptive), adding features (perfective) and preventing future problems (preventive). Deliverable: updated versions, patches and support.

Phase	Deliverable

1. Requirement Analysis	SRS document
2. Design	Design document
3. Implementation	Working code
4. Testing	Test reports
5. Deployment	Live software
6. Maintenance	Updates and patches

L6 Explain the four types of software maintenance with examples.

BOOK EXERCISE 4(V) LIKELY

After the software is released to users, it does not stop; it needs ongoing support and updates. This work is called software maintenance, and it is the last phase of SDLC. There are four types of maintenance.

1. CORRECTIVE MAINTENANCE

This type is used to fix the bugs and errors that are found by the users after the release. Example: fixing a login error, or fixing a printer driver problem that stops a report from printing.

2. ADAPTIVE MAINTENANCE

This type is used to change the software so that it keeps working in a new or changed environment, such as a new operating system, new hardware or new rules. Example: updating the software for a new version of Windows, or adding an e-book category to a library system.

3. PERFECTIVE MAINTENANCE

This type is used to improve the software or add new features to make it better, even if there is no error. Example: adding a dark mode, or making the search faster when the number of books grows.

4. PREVENTIVE MAINTENANCE

This type is done to prevent future problems before they happen, by improving the code or taking safety measures. Example: improving the code for better performance, or adding an automatic daily backup.

Type	Purpose	Example
Corrective	Fix bugs found after release	Fixing login error
Adaptive	Change for a new environment	Update for new Windows version
Perfective	Add features or improve	Add dark mode
Preventive	Prevent future problems	Better performance, automatic backup

L7 Explain the Waterfall model with its characteristics, advantages, disadvantages and when to use it.

BOOK TOPIC 1.2.4 LIKELY

SDLC models define how the phases of software development are organized. The Waterfall model is a linear and sequential approach. Each phase (Requirements, Design, Implementation, Testing, Deployment and Maintenance) must be completed before the next phase begins, and there is no going back, like water falling down steps.

CHARACTERISTICS

- It is sequential: one phase is completed after another with no overlapping.
- It needs heavy documentation, so every phase produces written records.
- Changes are difficult and expensive once a phase is finished.

ADVANTAGES

- Simple to understand: the process is clear and linear.
- Easy to manage: phases and milestones are defined.
- Well documented: complete records are kept at each phase.
- Works for small projects when the requirements do not change.

DISADVANTAGES

- Not flexible: it is hard to make changes after a phase is complete.
- Late testing: bugs are found late in the process, and fixing them is costly.
- High risk: problems discovered late can be very expensive.
- Not suitable for complex projects that have changing requirements.

WHEN TO USE THE WATERFALL MODEL

Use it when the requirements are clear and fixed, the project is small to medium in size, the technology is well understood and no major changes are expected.

CASE STUDY

The Al-Noor Library Management System was built with the Waterfall model in 6 months by a team of 4 members, because the school knew exactly what the system should do, the requirements were frozen after the SRS was signed, and the project had a fixed deadline.

L8 Explain the Agile model and the Scrum cycle. Write its advantages and disadvantages.

BOOK TOPIC 1.2.4 LIKELY

The Agile model is an iterative and incremental approach to software development. Instead of building the whole software in one long process, the work is divided into small cycles called sprints (usually 2 to 4 weeks), and a working part of the software is delivered at the end of every sprint.

CHARACTERISTICS

- Iterative and incremental: the product grows sprint by sprint.
- Flexible to changes: new requirements can be added at any time.
- Customer involvement throughout the project, not only at the start and the end.
- Working software is delivered frequently, and team collaboration is emphasized.

SCRUM CYCLE

Scrum is the most popular Agile framework. The cycle starts with the Product Backlog (the list of all features written as user stories). In Sprint Planning the team selects the stories for the sprint. During the 2-week Sprint the team works with a short Daily Standup meeting every day. At the end the Sprint Review shows the work to the client for feedback, and the Sprint Retrospective discusses how to improve. Working software is delivered after every sprint.

ADVANTAGES

- Flexible: it is easy to add or change features.
- Early delivery: users see working software quickly.
- Customer satisfaction: regular feedback and involvement.
- Lower risk and better quality: problems are found and fixed early through continuous testing.

DISADVANTAGES

- Less documentation, which may cause problems later.
- Requires commitment: the customer must be available.
- Unpredictable: it is hard to estimate time and cost.
- Scope creep: features may keep expanding.

WHEN TO USE THE AGILE MODEL

Use it when the requirements may change, the customer wants to see progress frequently, the project is complex or large, and innovation and speed are important.

CASE STUDY

The Al-Noor Student Mobile App was developed with Agile (Scrum) in 5 sprints of 2 weeks. The requirements were unclear at the start, so features such as the timetable, exam results, attendance notifications, library booking and homework submission were added sprint by sprint after feedback.

L9 Compare the Waterfall and Agile models using the two case studies of the book. When would you use each?

BOOK EXERCISE 3(III) LIKELY

The same school, Al-Noor High School, used two different SDLC models for two different projects, depending on the nature of each project. This shows that no model is best for every situation.

CASE STUDY 1: WATERFALL (LIBRARY MANAGEMENT SYSTEM)

The school knew exactly what the library system should do, the requirements were fixed and the deadline was clear. The team of 4 completed Requirements, Design, Coding, Testing, Deployment and Maintenance one after another in 6 months, and the complete system was delivered at the end.

CASE STUDY 2: AGILE (STUDENT MOBILE APPLICATION)

The requirements were not clear at the beginning, so students, teachers and parents asked for new features as the project went on. The team of 5 worked in 5 sprints of 2 weeks, and the first working version (basic login and timetable) was ready after the first sprint. Feedback was added in the next sprints.

COMPARISON TABLE

Dimension	Waterfall (Case Study 1)	Agile (Case Study 2)
Requirements	Fully defined and fixed	Partially defined, evolved over time
Client involvement	At the start and the end	After every sprint
First working product	After 6 months	After 2 weeks
Response to change	Difficult, needs to go back to Phase 1	Easy, added in the next sprint
Documentation	Heavy (SRS, design, test report)	Light, working software is preferred
Testing	One dedicated phase after coding	Continuous, in every sprint
Risk	Higher, late bugs are expensive	Lower, problems found early

WHEN TO USE WHICH

Use Waterfall when the requirements are fixed and fully understood at the start, and use Agile when the requirements are unclear or are expected to change.

L10 What is a communication model? Explain the seven layers of the OSI model in detail.

BOOK EXERCISE 4(IV) LIKELY

A communication model is a set of rules and layers that explains how data travels from one device to another over a network. It is like the rules of a postal system: both the sender and the receiver must follow the same steps for the letter to arrive. The OSI (Open Systems Interconnection) model was developed in 1984 by the International Organization for Standardization (ISO). It divides network communication into seven layers; each layer has a specific job and communicates only with the layer directly above and below it. Mnemonic from the bottom: "Please Do Not Throw Sausage Pizza Away".

LAYER 7: APPLICATION LAYER

It is the topmost layer and is closest to the user. It provides network services to the user's applications such as a web browser or an email client. Protocols: HTTP, HTTPS, FTP, SMTP, DNS. Analogy: the menu in a restaurant.

LAYER 6: PRESENTATION LAYER

It acts as a translator for the network. It converts data formats (for example Unicode to ASCII), encrypts and decrypts data for security, and compresses data to make it smaller. Protocols and standards: SSL/TLS, JPEG, MP3. Analogy: a translator at an international meeting.

LAYER 5: SESSION LAYER

It sets up, maintains and closes the communication session between two computers, and uses checkpoints so that a transfer can resume after an interruption. Protocols: NetBIOS, RPC, PPTP. Analogy: a telephone call, from "Hello" to "Goodbye".

LAYER 4: TRANSPORT LAYER

It provides reliable end-to-end communication. It breaks data into segments, reassembles them in the correct order, does error control and flow control, and uses port numbers. Protocols: TCP (reliable) and UDP (fast). Analogy: a courier company that guarantees delivery.

LAYER 3: NETWORK LAYER

It does logical addressing (IP addresses) and routing, which means finding the best path for data across networks. Routers work at this layer. Protocols: IP (IPv4, IPv6), ICMP, RIP, OSPF. Analogy: GPS navigation.

LAYER 2: DATA LINK LAYER

It organizes bits into frames, adds physical (MAC) addresses and detects errors between two directly connected devices. Switches work at this layer. Technologies: Ethernet, Wi-Fi and PPP. Analogy: a local postal worker who uses house numbers.

LAYER 1: PHYSICAL LAYER

It is the lowest layer and sends the raw bits (0s and 1s) as electrical signals, light pulses or radio waves over the physical medium. Hardware: cables, hubs, repeaters and network interface cards. Analogy: the roads and highways that vehicles travel on.

Layer	Name	Data unit	Device or example
7	Application	Data	Browser, email client
6	Presentation	Data	Encryption, compression
5	Session	Data	Session control
4	Transport	Segment	TCP, UDP
3	Network	Packet	Router
2	Data Link	Frame	Switch
1	Physical	Bits	Cable, hub

L11 Explain the TCP/IP model, its four layers and how they map to the OSI model.

BOOK TOPIC 1.3.5 LIKELY

The TCP/IP model (Transmission Control Protocol / Internet Protocol) is the practical model that the Internet actually uses. It was developed in the 1970s by the United States Department of Defense. Every device that connects to the Internet, such as a computer, a smartphone or a smart television, uses TCP/IP in the background every time we visit a website, send an email or make a video call. It has four layers.

1. APPLICATION LAYER

It combines the functions of the OSI Application, Presentation and Session layers into one layer. It gives all the services that user-facing applications need, including data formatting, encryption, session management and the actual application protocols. Protocols: HTTP, HTTPS, FTP, SMTP, DNS, IMAP and POP3.

2. TRANSPORT LAYER

It is identical in function to the OSI Transport layer (layer 4). It provides end-to-end communication using TCP for reliable, connection-oriented delivery and UDP for fast, connectionless delivery. It uses port numbers to identify applications.

3. INTERNET LAYER

It corresponds to the OSI Network layer (layer 3). It is responsible for logical (IP) addressing and for routing packets across multiple networks. Routers operate at this layer. Protocols: IPv4, IPv6, ICMP, ARP, OSPF and BGP.

4. NETWORK ACCESS LAYER

It is also called the Link layer and it combines the OSI Data Link and Physical layers. It handles everything related to the local physical network: MAC addressing, framing, error detection and the actual transmission of bits over cables or wireless signals. Technologies: Ethernet, Wi-Fi and cables.

TCP/IP layer	Corresponding OSI layers	Key protocols

Application	Application, Presentation, Session (7, 6, 5)	HTTP, HTTPS, FTP, SMTP, DNS
Transport	Transport (4)	TCP, UDP
Internet	Network (3)	IPv4, IPv6, ICMP, ARP
Network Access	Data Link, Physical (2, 1)	Ethernet, Wi-Fi, MAC

CONCLUSION

Both models share the same Transport and Network (Internet) concepts. The main difference is that TCP/IP puts the top three OSI layers into one Application layer and the bottom two OSI layers into one Network Access layer, which makes it simpler and more practical.