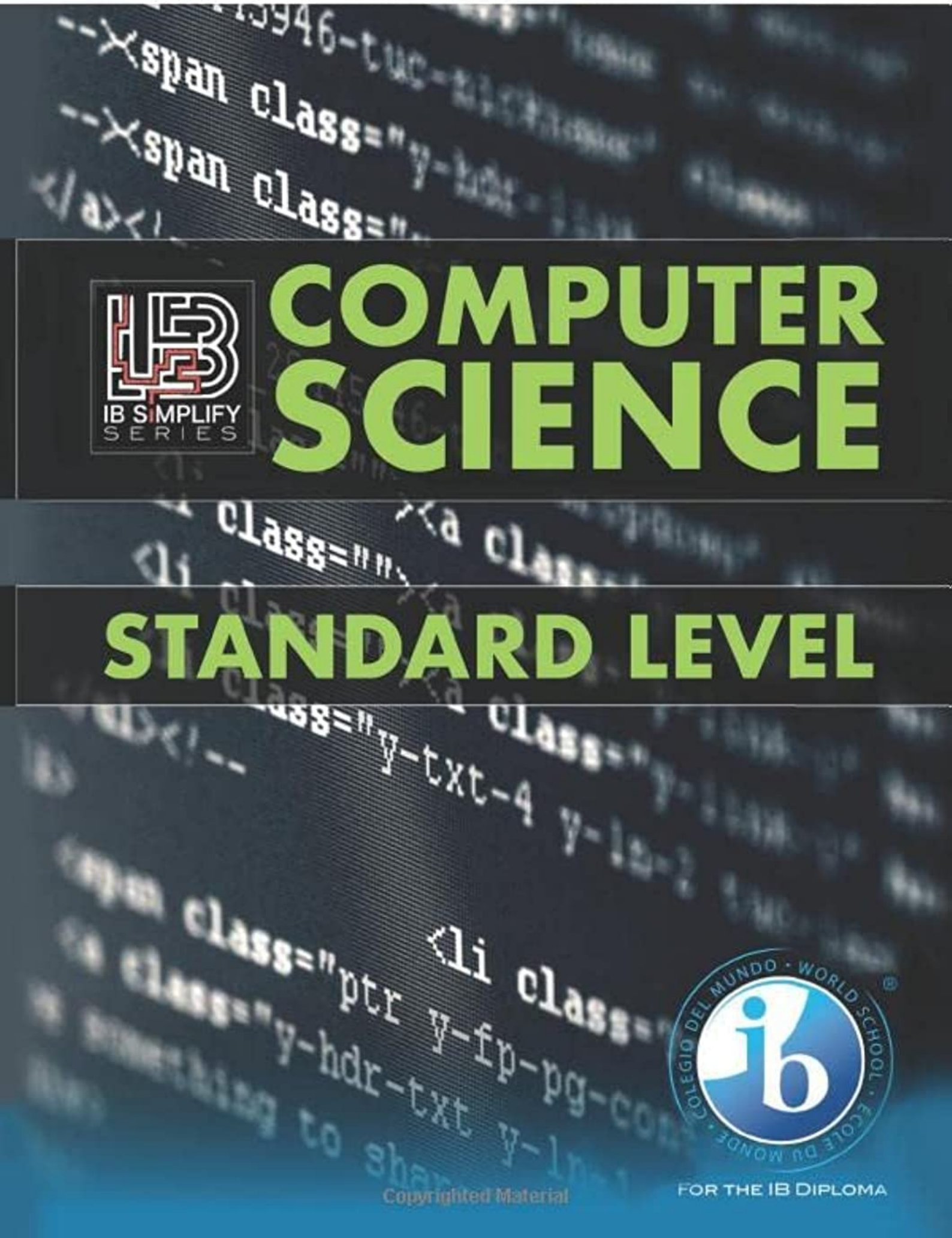
STUDY AND REVISION GUIDE



IB SIMPLIFY SERIES

COMPUTER SCIENCE

STANDARD LEVEL

FOR THE IB DIPLOMA

TABLE OF CONTENTS

Exam Questions by Topic..... 1
Section 1: SYSTEM FUNDAMENTALS..... 2
(1.1) Planning and System Installation 2
(1.1.1) Reasons for a New System..... 2
(1.1.2) Change Management..... 4
(1.1.3) Compatibility Issues..... 4
(1.1.4) Locally .vs. Remotely Hosted Systems..... 6
(1.1.5) Installation Processes..... 7
(1.1.6) Problems with Data Migration..... 9
(1.1.7) Types of Testing..... 10
(1.2) User Focus..... 12
(1.1.8) User Documentation..... 12
(1.1.9) Methods of User Documentation..... 12
(1.1.10) User Training Methods..... 14
(1.3) System Backup..... 15
(1.1.11) Causes of Data Loss..... 15
(1.1.12) Consequences of Data Loss..... 15
(1.1.13) Measures to Prevent Data Loss..... 16
(1.4) Software Deployment..... 18
(1.1.14) Types of Updates..... 18
(1.5) Computer Components..... 20
(1.2.2) Roles of a Computer in a Networked World..... 20
(1.2.3) Ethical and Social Issues with the Networked World..... 22
(1.6) System Design and Analysis..... 25
(1.2.5) Obtaining and Evaluating Requirements..... 25
(1.2.8) Illustrating System Requirements 26
(1.2.9/10) Iteration (Iterative Design)..... 27
(1.2.12) Human Interaction with the System..... 27
(1.2.13) Usability Problems and Examples..... 27
Section 2: COMPUTER ORGANISATION..... 30
(2.1) Computer Architecture..... 30
(2.1.2) Primary Memory..... 31
(2.1.3) Cache Memory..... 31
(2.1.1) Control Unit (CU) Registers..... 32
(2.1.4) Machine Instruction Cycle..... 33
(2.1.5) Secondary Memory..... 34
(2.1.6) Operating and Application System..... 35
(2.1.7) Software Application..... 37
(2.1.8) Common Features of Applications..... 38

Section 3: NETWORKS..... 40
(3.1) Network Fundamentals..... 40
(3.1.1) Type of Networks..... 41
(3.1.5) Virtual Private Network (VPN)..... 45
(3.1.4) Technology Required to Provide (VPN)..... 41
(3.1.2) Importance of Standards in Construction of Network..... 47
(3.1.3) Network Communication Layers..... 47
(3.2) Data Transmission..... 48
(3.1.11)Data Switching..... 48
(3.1.7) Protocols..... 49
(3.1.9) Data Compression..... 51
(3.4) Network Security..... 53
(3.1.15) Symmetric-Key Encryption..... 53
(3.1.16) Public Key Encryption..... 53
Section 4: APPLICATION OF COMPUTER SCIENCE..... 56
(4.1) Binary Representation..... 56
(2.1.10) Binary Representation..... 57
(2.1.11) Boolean Logic..... 59
(4.1.14-16) Concurrent Processing..... 61
(4.1.17) Abstraction..... 63
(4.2.1) Standard Algorithms on Linear Arrays..... 63
(4.3.5) Higher Level Language and Machine Code..... 65
(4.3.6) Use of Programming Language..... 66
(4.3.12) Discuss the Need for Sub-Programmes and Collections..... 66
(4.2.2) Standard Operations of Collections..... 67
(D1.1/2) Classes and Objects..... 67
(D1. 3/4) UML Diagrams..... 69
(D1.5) Decompostion into Related Objects..... 70
(D1.6) Describe the Relationship between Objects..... 71
(D1.7) Outline the Need to Reduce Dependencies between Objects..... 72
(D1.10) Describe How Data can be Pass to/from Actions as Parameters..... 72
(D2.7) Advantages of Libraries of Objects..... 72
(D2.8) Describe the Disadvantages of Object-Oriented Programming..... 72
(D2.9/10) Discuss the Use of Programming Teams..... 73
(D2.8) Describe the Disadvantages of Object-Oriented Programming..... 73
(D3.1-3) Definition of Terms..... 73
(D3.9/10) Modern Programming and Ethical Considerations..... 75
(4.3) Java Object Oriented Programming..... 76
(D1.9) Primitive Data Types..... 80

EXAM QUESTIONS

TOPIC 1: SYSTEM FUNDAMENTALS(QUESTIONS)

1.1	Identify two features that need to be considered when planning a new computing system for an organisation.	2
1.1.2	(b) Other than data migration, describe two aspects of change management that may arise from this takeover.	4
1.1.3	Outline one problem of maintaining legacy systems.	2
1.1.3	(c) (i) Identify one external threat to the security of the school's computer system.	1
1.1.3	(ii) State one way to protect the computer system from the threat identified in part (c)(i).	1
1.1.4	Describe one way that software developers can ensure that the users are aware of any available updates for their products.	3
1.1.4	(a) Describe the features of SaaS.	3
1.1.5	(e) Outline one economic aspect that the examination office needs to take into account to support parallel running.	3
1.1.5	(i) Outline what is meant by parallel running.	2
1.1.5	(ii) Outline one reason for choosing parallel running as opposed to a direct changeover.	2
1.1.5	(g) Compare direct changeover and phased conversion.	4
1.1.6	(a) Identify two aspects of the data that need to be taken into account during the planning of the new system.	2
1.1.6	(d) Discuss two possible problems that may occur during data migration.	4
1.1.6	(c) Outline two problems that may occur when transferring data from the old system to the new system.	4
1.1.6	(f) Outline one problem that may arise from the installation of new hardware and software in the office.	2
1.1.6	(a) (i) Define the term data migration.	1
1.1.6	(ii) Describe two problems, concerning data migration, which the company may have to overcome.	4
1.1.6	(d) (i) Identify two sources of risk to personal data in this online system.	2

1.1.6	(iii) Outline the consequences to the customer if their data is not adequately protected.	2
1.1.7	(c) Discuss whether beta testing would be appropriate in this scenario.	3
1.1.7	Explain what is meant by beta testing.	2
1.1.7	Describe the use of beta testing.	4
1.1.7	Explain what is meant by user acceptance testing.	2
1.1.8	(a) Describe two types of documentation that should be provided with the software package.	4
1.1.8	(c) State one method of providing user documentation.	1
1.1.8	Discuss one advantage and one disadvantage of printed material, when compared to online support, as a method to provide user documentation.	2
1.1.10	(b) State two methods of delivering user training.	2
1.1.11	(c) Identify three causes of data loss.	3
1.1.11	Identify two causes of data loss.	2
1.1.12	(c) Outline the consequences of data loss to customers and to the company.	2
1.1.13	(d) Describe one method that the company could use to prevent data loss.	3
1.1.13	(d) Describe why data loss is a more serious problem than the loss of software or hardware for a sales company.	3
1.1.13	(e) Identify two methods of preventing data loss.	2
1.1.13	(e) Outline the security measures that should be taken to prevent data loss.	2
1.1.13	(b) Describe two different methods that the medical centre could use that would allow data to be restored should it be lost for any reason.	4
1.1.13	Outline the use of a failover system.	2
1.1.14	Identify two reasons for releasing a software update.	2
1.2.3	(c) Discuss the issues that should be considered before making these decisions. (Data security)	6
1.2.3	(b) Discuss the limitations of SaaS in relation to security.	6

1.2.4	(i) Identify two different types of users of the system.	2
1.2.5	(b) Describe how direct observations on the current system may provide information to help propose a suitable new system.	3
1.2.5	(a) Describe one method by which systems requirements can be obtained from the stakeholders.	2
1.2.8	(c) Describe the purpose of this prototype.	3
1.2.8	(b) Describe why it is useful to produce more than one prototype of the new system.	2
1.2.8	(b) Outline one reason for providing a prototype for this new system.	2
1.2.8	(a) Construct a system flow chart for the system described above.	5
1.2.10	(ii) Explain the role of users in the process of developing the new computer system.	3
1.2.12	(a) Define the term usability.	1
1.2.13	Outline two usability features in relation to the characteristics of a new laptop.	4
1.2.13	Identify one method of inputting data that can improve the accessibility of a computer system for some users.	1
1.2.13	State three potential usability issues with cell phones.	3
1.2.14	(b) Identify two methods that could be used to improve the accessibility of a computer system.	2

TOPIC 2: COMPUTER ORGANISATION (QUESTIONS)

2.1.1	Outline the role of the memory data register in the machine execution cycle.	2
2.1.1	Describe the function of the control unit (CU) in the central processing unit (CPU).	2
2.1.1	(c) Outline the function of the: (i) memory address register (ii) memory data register.	4
2.1.1	(ii) Identify the part of the CPU which performs decoding.	1
2.1.1	Outline the function of the: (i) ALU (ii) CU	2
2.1.2	(a) State where all instructions and data are stored.	1
2.1.2	Distinguish between the use of two types of primary memory.	2
2.1.3	(a) State the purpose of cache memory.	1
2.1.3	Describe how the cache memory can speed up the functioning of a processor.	2
2.1.3	Explain why cache memory can speed up the processing within a computer.	3
2.1.4	(b) Outline the role of the data bus and address bus in this process.	2
2.1.6	(b) Outline the purpose of one other possible peripheral device in this scenario.	2
2.1.6	Outline one feature of the operating system that needs to be considered when running a game application.	2
2.1.6	(i) Identify two functions of the operating system.	1
2.1.6	Describe how this function prevents the system from crashing when more than one program is run at the same time. (OS Functions)	2
2.1.6	(ii) State where the operating system is held when the computer is turned off.	1
2.1.6	Explain the importance of the memory management function of an operating system.	3
2.1.7	(c) Identify two additional features of a word processing package that could be useful for this office	2

2.1.7	(d) Outline the purpose of one application software package other than a word processing package that could be used in this office.	2
2.1.7	Outline, with an example, one benefit of using computer-aided design (CAD) applications.	2
2.1.7	Outline two characteristics of spreadsheets.	2
2.1.8	(b) Identify two features of a user interface that will allow application programmers to interact more easily with the programming language.	2
2.1.9	Define the term bit.	1
2.1.9	In an 8-bit register, state the binary representation of the hexadecimal number 3B.	2
2.1.9	Calculate how many different colours can be represented using two hexadecimal characters.	2
2.1.9	(a) Calculate the number of bits in each memory location.	1
2.1.9	(b) Calculate the number of bytes in each address.	1
2.1.9	Outline the relationship between binary and hexadecimal.	2
2.1.10	Outline how a colour can be represented in a computer.	2
2.1.12	Construct a truth table for the Boolean expression: NOT (A XOR B) AND C	4
2.1.12	Construct a truth table for the Boolean expression: NOT (A XOR B) AND C	4
2.1.12	(c) (i) Contruct a truth table corresponding to this diagram.	3
2.1.12	(d) Outline how truth tables can be used to test that any two logic diagrams are equivalent.	2
2.1.13	Construct a logic diagram for the Boolean expression: NOT A OR (A AND B)	3
2.1.13	Construct a logic diagram for the Boolean expression: NOT A OR B AND C	3
2.1.13	Construct a truth table for the expression: A XOR (B OR C)	3
2.1.13	Construct a truth table for the following Boolean expression: (A AND B) NOR C	3
2.1.13	Copy and complete the following truth table:	3
2.1.13	(ii) Identify the single logic gate that is equivalent to this diagram.	1

2.1.13	Construct a truth table for the following Boolean expression: (A OR B) AND (NOT C OR B)	4
2.1.13	Construct the truth table for the following Boolean expression: X = NOT A AND B OR A AND NOT B	4

TOPIC 3: NETWORKS (QUESTIONS)

3.1.1	(b) Outline two disadvantages of this WLAN.	4
3.1.1	(d) Outline two advantages, to this company, of installing a WLAN.	4
3.1.1	(a) Identify one hardware component of the WLAN, other than computers.	1
3.1.1	In the context of the networked world, state the role of: (b) a server.	1
3.1.1	(e) Discuss any two of these issues and the ways in which the company might resolve them. (WLAN security issues)	4
3.1.1	In the context of the networked world, state the role of: (a) a client.	1
3.1.1	(a) (i) Identify two different types of network that can be combined to use the internet to address the above requirements.	2
3.1.1	(a) Define the term server.	1
3.1.1	(c) Identify three ways in which the network administrator can reduce the risk of unauthorized access to confidential data.	3
3.1.1	(b) Identify the different clients in this network. (LAN)	1
3.1.1	(ii) For one of the networks identified in part (a)(i), describe the security characteristics that are particular to this network.	2
3.1.1	(c) Outline one advantage and one disadvantage of allowing wireless access to the server. (LAN)	4
3.1.1	Identify a type of network that would allow secure access from an employee’s home to their company’s LAN.	1
3.1.1	Identify two key features of a peer-to-peer (P2P) network.	2
3.1.1	Explain why the speed of data transmission across a network can vary.	3
3.1.1	(c) Define the term extranet.	2

3.1.3	Identify any two of the layers of the OSI model.	2
3.1.3	(a) Outline the concept of the Open Systems Interconnection (OSI) model in communication across a network.	3
3.1.4	(b) Describe two features of a VPN that make it secure.	4
3.1.4	(c) State one technology that is necessary for a VPN.	1
3.1.4	Explain how a virtual private network (VPN) allows a travelling salesperson to connect securely to their company's network.	4
3.1.4	(d) Distinguis between a VPN and an extranet	4
3.1.5	Outline the features of a virtual private network (VPN)	2
3.1.5	(f) Explain how setting up a virtual private network (VPN) would provide a suitable solution.	4
3.1.5	Outline one example of the use of a virtual private network (VPN).	3
3.1.6	Identify two characteristics of a data packet.	2
3.1.6	Define the term protocol.	1
3.1.6	Explain why protocols are used in network communications.	3
3.1.6	(a) Define the term protocol.	1
3.1.6	(b) Outline why protocols are nessecary.	2
3.1.6	Define the term data packet.	2
3.1.7	(b) Discuss the importance of protocols in ensuring the successful preparation, transmission and delivery of data using packet switching.	6
3.1.7	(c) Outline the purpose of protocols in transferring this data.	2
3.1.7	(b) Outline, with an example, the function of protocols.	2

3.1.9	Explain how compression of data may lead to negative consequences.	3
3.1.11	With reference to Figure 1, explain how data is transferred by packet switching.	6
3.1.12	(c) Describe two advantages to society of the increased availability of WiFi outside the home.	4
3.1.12	Outline one advantage and one disadvantage of wireless networks.	4
3.1.13	(a) In the given context, distinguish between Ethernet and wireless in terms of reliability of transmission.	4
3.1.15	Explain how the use of media access control (MAC) addresses can improve security	3

Section 1

SYSTEM FUNDAMENTALS

1.1 Planning and System Installation

Reasons for a New System (1.1.1)

Revised

- The Old system is inefficient.
- The Old system is no longer suitable for its original purpose or is outdated.
- To increase productivity, efficiency and quality of output.
- To decrease system flaws and minimise costs.

The length of time required to implement a New System depends on hardware and software costs, people needed to develop the system and the immediate environment.

Employees may require training to use the system, and other employees may become redundant under the New System.

To decide if a System Update is worth pursuing, such issues should be considered:

- Compatibility between Old and New System.
- Strategies for Merging Systems.
- Data Migration.
- Hosting Systems.
- Installation Processes.

Features which need to be Considered when Planning a New System:

- Roles and activities of the users (e.g. Permissions, Security, Partitions, Collaborative Work).
- Resources (Hardware and Software equipment) appropriate fort the organisation.
- Costs/budget limits.
- Delivery time.
- Compatibility with the old system (Data).

One approach is to conduct a **feasibility report** which evaluates and analyses a project and its potential based on the follow criteria:

T **Technical Feasibility**

Is the existing technology sufficient to implement the proposed system?

E **Economic Feasibility**

Is the proposed system cost effective?

L **Legal Feasibility**

Are there any conflicts between the proposed system and any regulations?

O **Operational Feasibility**

Are the existing organisational practices and procedures sufficient to support the maintenance and operation of a new system?

S **Schedule Feasibility**

How long will it take to implement?

Change Management (1.1.2)

Revised ☐

Management of people, departments and organisations from one state of a previous System into the desired New System.

Successful Change Management should maximise benefits and minimise impact of change on people so that stakeholders accept the change in the environment; also, issues regarding planning the system should be resolved.

Issues with Change Management:

- Workforce issues such as redundancy/retraining.
- Time frame involved in merging the two systems.
- Testing of the combined systems/new data.
- Data entry if migration is not possible.
- Costs involved in the aligning of the two systems.
- Changeover decisions such as parallel running, etc.

Compatibility Issues (1.1.3)

Revised ☐

Business Mergers:	When two businesses combine, it is important to ensure that their operational systems are compatible.

Issues with Merging Businesses:

- **Language Differences;**
  Communication issues and different interpretations.
- **Software Incompatibility;**
  Different software systems cannot operate well on the same computer or network.
- **Legacy Systems;**
  The term Legacy System refers to Old technology, hardware, software, computer systems or programs.

Some Legacy Systems satisfy current user needs; Data cannot be converted into newer formats; or Applications processing data are Upgraded — so Legacy Systems remain in place even though newer technology is available.

Issues with Maintaining a Legacy System:

- Expensive to change since programs may be disorganised, and documentation may be missing.
- Could cause compatibility issues (Often due to Old Programming Languages or Old Database Technology).
- May be difficult to recruit staff/programmers familiar with old languages/operating systems.
- Typically pre-internet and need an interface program to interact with the system.

  The system mainframe is usually large and complex. Even maintaining the system may lead to security vulnerabilities due to system incompatibility.

Strategies for Merging:

- Keep both systems, and develop them to have the same functionality.

  High maintenance cost.

  Consideration of the time frame this will take place within.

- Replace both Systems with a New System.

  High initial cost.

  Some employees will be Fired or Retrained.

- Combine the Best Systems from both companies.

  Difficult for employees to work with different Systems.

  Requires Testing.

- Only use one System.

  Policy and Employee issues.

Locally .vs. Remotely Hosted Systems (1.1.4)

Revised ☐

Locally Hosted System: Software installed and operated on the clients own hardware infrastructure (e.g. Owning paid-for packaged software)

Reasons For:

- Best solution for large, complex Systems.
- Only paid for once, excluding maintenance costs (which would address updates to the software).
- Data is controlled in a secure data centre, which decreases the risk of data loss.

Reasons Against:

- Higher initial costs compared to Remote Hosting.
- Harder to predict the total cost, which could make it more expensive in the long-term with constant maintenance costs.
- System must be maintained yourself.

Remotely Hosted System: Software-as-a-service (SaaS) which means Hardware used to operate the System is elsewhere, with Data held in the Cloud.

Access to this software is through a Thin Client or via the Extranet.
Service is paid for on a Subscription basis.

Reasons For:

- Lower initial Cost.
- Overall cost is much more predictable (fixed to the Subscription fee).
- Best solution for organisations without their own hosting hardware.
- System maintenance is managed by the provider, not the client.
- Data is secured in a data centre.

Installation Processes (1.1.5)

Revised

Changeover The process of putting the New System online and retiring the Old System.

Parallel: Simultaneous operation of the New and Old System.
Both Systems run Parallel to each other at first to compare outputs until satisfied with the New System, then the Old System is retired.

Reasons For:

- Less risk, since if the New System fails, the Old System can be implemented again.
- Safe way of Validating and Evaluating the New System without any consequences if the New System fails.
- Running Two Systems could be cheaper than losing all the data in case of failure.

Reasons Against:

- Higher Cost; Simultaneous Operation is very costly since both systems and all their Resources (such as staffing requirements) must be maintained.
- Inefficient if systems have different Processing Tasks (inputs, outputs, functions).
- Employees may be trained for a New System only to use the Old System.

Direct: Also known as 'Big Bang'. New System set up and Old System terminated at the same time. Preferred if the data or system is not critical.

The old software and hardware is completely replaced, in one move, by the new software and hardware. Direct changeover means everyone in the organisation has the same software/hardware.

Reasons For:

- Less costly.
- No compatibility issues; Since all users have the same Software and Hardware.

Reasons Against:

- Higher risk of data loss and lack of functionality.
- Preferred when the System is not critical.
- Cost of training users before the changeover takes place, so that they can use the New System efficiently.

Pilot: Used in organisations with multiple sites. The New System is introduced to one of the sites (the pilot site) and then introduced to others if successful.

Reasons For:

- Less risk.
- Pilot site (First group that adopts the System) can serve as a model for the rest of the System.

Reasons Against:

- Extensive worker training required.
- Longer Changeover period.

Phased: Selects one area where the changeover will take place for the New System, then when this area is running satisfactorily, other areas will be changed to the New System.

Reasons For:

- Less risk as any problems that might arise will be isolated in only one area of the System.

Reasons Against:

- Training period and implementation takes much longer.

Problems with Data Migration (1.1.6)

Revised

Data Migration: Transferring data between formats, storage types and computer systems when switching to another System.

Data Migration Risks:

- **Incompatibility of Type and Format**

  It is necessary to translate Data from one Format to another, otherwise incompatibility may cause performance issues/delays in System operation.

  File formats in the New System may not be compatible and could lead to incomplete data transfer. (e.g. the data Records in databases may have different Structure (fields) or the data.

  Files could be incompatible if the data is from different Hardware systems).

- **Data Loss or Corruption**

  When moving data from one storage device to another (via network, cables etc), data could be corrupted or lost and consequently no longer useful. Transmission faults or lack of adequate storage may result in unusable data or total data loss at its destination.

- **Frequency of Access**

  Some data is not frequently accessed and so can be 'archived' (transfer to an infrequently used storage medium such as Magnetic Tape).

- **Quantity and Size of Data**

  Data should not exceed the storage capacity of the New System.

- **Different Validation Rules**

  This difference between company data validation protocol could lead to inconsistent or incorrect data results. Data may also be unusable during transfer.

- **Type of Access: Access Right Permissions**

  Some users will have different access permissions than others, for example Students may have read only access, whereas Teachers may have the ability to read sensitive data and write files.

  Information could be sold or stolen for Fraudulent purposes (e.g. Junk Mail) or Personal details stolen (e.g. Use in Identity Theft).

- **Employee Issues**

  Employees may be afraid of System changes for whatever reason, and unwilling to participate in the transfer. Furthermore, Employee efficiency may drop as they learn to adopt the New System.

Types of Testing (1.1.7)

Revised

Testing is essential since it identifies Problems to be fixed, areas for Improvement and determines whether the System or Software fulfils Requirement.

If not done properly, an inadequate system means inadequate employee productivity, reduced efficiency and output, increased costs and end user dissatisfaction.

Alpha Testing	Offering early development versions to other developers before available to the general, allows for feedback.
Beta Testing	Testing prior to the Release. Performed by end-users (not by designers) outside of Development to receive real-world Feedback. The involvement of other uses can increase the objectivity of Evaluation; However, user reports are not of technical quality, and there are many reports of the same bugs.
User Acceptance Testing	Usually the Final Stage, provided to clients as a last-minute check that the product satisfies the Target user.

Debugging	Systematically finding and correcting bugs (errors) - can be automated.
Validation	The process of evaluating whether data input follows appropriate specifications and is within reasonable limits.
Verification	The process of ensuring that the data input is the same as the original source data. One way verifying input is through 'Double Entry' or Data (inputting data twice in the same way).

1.2 User Focus

User Documentation (1.1.8)

Revised ☐

Allows users to quickly adapt to the software or system changed with minimal costs or inefficiencies. Internal Documentation is the Code Comprehension features used by developers to understand the technical aspects of the code itself. External Documentation is typically written as a separate document, providing software requirements, support documentation and description.

User Documentation typically includes:

- Requirements; Identifies attributes and functions of the System.
- Technical; Provides details on how to install and configure the System.
- End-User; Manuals for the end-user, software support staff and system administrators, with details on how the System is used.

Methods of User Documentation (1.1.9)

Revised ☐

Help Files	Easy to access, cheap, cannot be misplaced and no internet connection required.
Online Documentation	Easier to use and search through, can have forum or email support and can update the documentation (e.g. Email Support, Online Portals, Live Chat). However, access is limited by internet connection.
Technical Documentation	Describes how to install software and the hardware configuration needed.
User Documentation	Describes various functions of the software and helps users to learn how to use the software.

Manuals

Printed manuals can be accessed at any time, even if the software/system is not installed, and is not restricted by internet access since it is a hard copy.

However, it cannot be updated, and digital manuals are more cost-effective, and better for the environment (but again limited by internet connection).

PRINTED MATERIAL .VS. ONLINE SUPPORT AS A METHOD TO PROVIDE USER DOCUMENTATION:

Advantages:

- Portability; Printed material is more easily transportable and can be moved around (e.g. for a scanner or printer).
- Extent of Material; Books and technical instructions may have more details and be more useful to provide deeper explanations.
- Availability; It is always available (there are no power cut issues);

Disadvantages:

- Readability; Font size online can easily be magnified;
- Trouble shooting/Cross-Reference; Online is faster and usually has links to other pages whereas paper is a thick manual.
- Ageing; Online is more frequently updated than paper manuals;
- Environmental; Waste of paper versus energy consumption;

Role of Users in Development

- The role of Users is important because inadequate user involvement leads to project failure.
- All users must participate and explain how they use the system, using this opportunity to explain what they believe is wrong with the Current System and could be improved in the New System.
- Users (managers and owners) are involved in approval of projects and budgets.
- All users are involved in testing of the system and in training to learn how the New System functions.

User Training Methods (1.1.10)

Revised ☐

Self-Instruction: People can use resources such as manuals, websites, video tutorials to learn on their own.

Reasons For:

- Easiest and cheapest with more flexible time for the user.
- No tuition fee for the Student.

Reasons Against:

- Typically only used for easy-to-use or common use programs.
- Lack of user Guidance.
- Requires sufficient documentation to aid with training.
- Effectiveness depends heavily on users' motivation and ability to work on their own.

Formal Classes: Classroom setting with free discussion.

Reasons For:

- Students can exchange ideas and have direct interaction with an expert (teacher).

Reasons Against:

- Some members may work better/more productively on their own.
- Self-assured students dominate class discussion.

Remote/Online Training:

Reasons For:

- Larger variety of courses online.
- Accessible any time.
- Easy to set up and add new members, therefore cheaper.

Reasons Against:

- Excludes those without access to internet or IT skills required to use it.
- Might not be as effective for dependent learners who work best in groups.

In general, employees need to learn quickly and easily in order to implement the new system faster; therefore reducing costs and inefficiencies.

1.3 System Backup

Causes of Data Loss (1.1.11)

Revised ☐

User	Error Accidental deletion (file Recovery software needed);
	Administrative errors, saving in the incorrect location, Poor Data Organisation;
	Closing program without Saving (Autosave software required);
	Continued use after signs of Failure (Regular backups required).
Natural Disasters	Building Fires (need to Store Data in Two Locations);
	Floods, Earthquakes, Electrical Storms;
	Physical Damage to the Storage Device.
Malicious Activities	Someone purposefully deleting, altering or stealing data (by Disgruntles Employees or via external Hacking).
Corruption	Computer Viruses (need for Antivirus software);
	Errors in Computer Data (need for Backups);
	Firmware Corruption or Power Failure;
	Transmission Error (e.g. Lossy compression)

Consequences of Data Loss (1.1.12)

Revised ☐

These can be Serious and Fatal (e.g. Hospital data failure) and may have Enormous Costs (e.g. may have to repeat procedures, poor end-user/customer experience, cost of retrieving data).

Could possibly Lose Customers and severely damage company Reputation, or may Inconvenience users by failing to provide full functionality due to Data Loss.

Measures to Prevent Data Loss (1.1.13)

Revised

Regular Backups	On hard disks/magnetic tape, online or on 'removable storage media' such as USB or CDs for fast backup and storage.
Offsite Storage	Snapshots or Backups made on a regular basis; in the case of failure a dated/time-stamped copy exists and the state up until then can be used to restore customer records. Data backups stored in a different geographical location.
Firewall and Antivirus	Prevents virus infections. Using Passwords; Cryptography and Spyware Security Antivirus Programs; Prevention of physical damage to the computers; Keeping equipment in safe and dust-free places. Also Protect equipment from static electricity that can erase data or damage components.
Failover Systems	Computer system that can be switched to in case of hard/ software or network failure. Often switches automatically to reduce time. It is standby/redundant system which is used to eliminate/reduce the impact on users and owners by automatically taking over if the primary system suddenly becomes unavailable.
Use of a Mirror System	All changes to the records are made on two systems, if one or Log Files fails then the other holds all the current data. Also a log file which keeps a record of processes/transactions

THE SIGNIFICANCE OF DATA LOSS

Loss of Hardware should not be a major problem if its cost is not too Expensive, and in most cases the hardware supporting the System will be Insured (new substitute hardware could be found quickly and repurchased).

Loss of Software should not be a critical problem if the owner has made Backup copies, meaning the software could be Reinstalled.

Loss of Data could be very Expensive for the company and could also extend to Security Problems; misusing information about the company and individuals that are stored in the company's database. Data is also difficult and often impossible to Replace.

1.4 Software Deployment

It is important that users can install updates because otherwise they might not have fixes for Bugs or be able to benefit from added Features made to the system.

A software update may be released to:

- Patch any vulnerabilities/bugs.
- To provide improved functionality/new functions or to increase usability/efficiency.
- To generate income for the software company to innovate and stay ahead of other software companies.
- To ensure compatibility with other (updated) technologies.

When the software is Installed and Registered (a cookie is placed on the machine); This communicates with the software developer automatically on start up. Messages about updates are sent back to the machine and alerts are given. In other cases, an email will be sent with a link to the update.

Types of Updates (1.1.14)

Revised ☐

Patches	Used to fix known bugs and vulnerabilities. However, may introduce new bugs/errors as a side effect.
Updates	Used to fix bugs, add minor functionalities to the system, usually Free.
Upgrades	Used to fix bugs, add major functionalities or characteristics to the system, often required to be purchased.
Releases	Final, working applications gone through Alpha and Beta testing.

STRATEGIES FOR ALERTING USERS ABOUT UPDATES

Automatic

Cookie (software which monitors users activity) is integrated when software is registered and installed, this communicates with the developer automatically.

If an update if available for software, messages and alerts are returned.

Sending an Email

User registers email and other details when installing software.

Email then sent to the registered user with a link to download the update.

1.5 Computer Components

Computer Hardware (1.2.1)

Revised

Physical, tangible elements of a computer system (e.g. CPU, HDD).

Software — Set of instructions for the CPU to perform specific operations, comprised of both programs and data.

Peripheral Device — Auxiliary device that can connect to, communicate and work with the computer (e.g. input/output devices, printers, etc).

Computer Network — A set of computer systems that are interconnected and share resources, as well as data.

Human Resources — The set of individuals who make up the workforce of an organisation, business sector or economy. There is software that combines human resources functions (e.g. payroll, recruiting and training, performance analysis) into one package.

Roles of a Computer in a Networked World (1.2.2)

Revised

Dumb Terminal — A device that usually consists of a keyboard, monitor and network card which is connected to a server or powerful computer. Dumb Terminals depend entirely on the computer to which they are connected to for any processes.

Thin Client — A relatively low performance terminal, which heavily depends on the server to which it is connected.

Client	Receives data via the network, whereas the server has saved data on it and offers this to clients. It accesses services made available by server, by sending requests to the server.
Server	Program/host computer that fulfils requests from client programs or computers across network and shares info to clients.
Email Server	Message transfer agent that transfers electronic messages from one Computer to another in a network (provides access to email services, like a digital Post Office).
DNS (Domain Name Server)	Server that translates web addresses written in letters (more memorable for humans) to the numeric IP (Internet Protocol) address.
Router	Connects Networks together to forward Data Packets between networks, deciding where to send information from its destination address and routing policy.
Domain Name System Server (DNS)	Attributes names to network addresses and therefore allows users to input String names instead of numerical IP address.
Firewall	Controls incoming and outgoing network traffic, determining what data packets should be allowed through, based on a rule set. Needed to protect the integrity of client computer.

Ethical and Social Issues with the Networked World (1.2.3)

Revised

Security	Protecting Hardware/Software, peripherals, data and networks from unauthorised access. Access Levels, which control an individuals access permission to sensitive data can be put in place with strong Authentication Systems (e.g. Biometrics, Passwords, Firewalls, etc) in place. Cloud Storage may effect the security of the data. If data is sold to other providers for information purposes, this could seriously compromise clients. Different Data Legislation laws between countries may increase security risk to those using a Remotely Hosted System.
Privacy	Controlling how and to what extent data is accessed and used by others, to protect identity (e.g. GPS services on phones, data sold to third parties). However, also problems with anonymity (e.g. cyber bullying, hacking, terrorism, etc).
Censorship	Some information may be deemed inappropriate. Network manager could make sure no other computers can access it (e.g. China's Facebook block).
People and Machines	Easier communication, more information and efficiency. However, issues with addiction, neglect of reality, lack of sleep, health issues, car accidents, technical unemployment (work replaced by technology), digital alterations (fake news).
Digital Divide and Equality of Access	Inequalities regarding use and access to computer systems in different environments, leading to the inequality in information and education access.

Surveillance	Monitoring people (e.g. for law enforcement, conduct of employees, traffic control, etc), ethics of privacy and consent to surveillance need to be considered.
Globalisation and Cultural Diversity	Spread information and reduce political, geographical and financial boundaries. However, may diminish traditional cultures.

EMPLOYEES WORKING FROM HOME:

Reasons For:

- Saving in fuel cost and commuting time;
- An opportunity to work at your own pace (more comfortable, therefore increased productivity); Able to work in an undisturbed environment, Choose their own work hours.
- Convenience of not having to travel.

Reasons Against:

- Employers can clam the employees are part time consultants or the like to avoid paying benefits like insurance, medical plans, taxes;
- Stain on families that result when a family member works from home;
- At-home employees miss interaction with co-workers at the office;
- At-home employees think they work too much, employers think they do not work enough;
- Employees cannot monitor employees; Expensive of setting up a VPN.

PRACTICAL ISSUES TO CONSIDER WHEN NETWORKING

Reliability	How consistently a computer system functions according to its specifications with minimal system failure. Failures result in data, time or revenue loss.

Data Integrity and Consistency	Maintenance of accuracy and consistency of data. Must be complete, up-to-date and unaltered. Inconsistent if theres data duplication/redundancy.
Standards and Protocols	Rules followed in development of systems, including proprietary standards (e.g. computers compatible with Microsoft operating systems), industry standards (formally decided, e.g. USB) and 'de facto' standards (e.g. 'QWERTY' keyboard).

1.6 System Design and Analysis

Stakeholder Has an interest or investment in a project and is impacted by how it turns out.

System analysts have to collaborate with all stakeholders: clients and end-users.

Obtaining and Evaluating Requirements (1.2.5)

Revised

Interviews: Face-to-face with verbal responses. Can be structured with the same questions and manner for every stakeholder, or unstructured with more flexibility.

Reasons For:

- Talk directly to users/members of the organisation and can observe non-verbal behaviour, meaning data can be considered more reliable and valid.
- Unstructured interviews can reveal more questions that otherwise wouldn't have been addressed, meaning more detailed reports.

Reasons Against:

- Data from unstructured interviews is hard to summarise, evaluate and analyse clearly.
- Level of detail depends on the type of interview - structured interviews get less detailed responses.
- Time-consuming to gather detailed results.

Questionnaires: Can be closed/restricted (box checking, yes/no form) or open/unrestricted (free response questions).

Reasons For:

- Time-saving and cost-efficient: can acquire information from a large group of people easily and cheaply.
- Closed/restricted questions makes data easier to compare.
- Open questions means more detailed reports can be created.

Reasons Against:

- Level of detail depends on type of questions: closed questions don't allow for clarifications or extra detail.
- Stakeholder could interpret the meaning of the question incorrectly, and therefore invalidate their answers.

Direct Observation of Current Procedures:	On-site observation of different departments to see where things c can be more efficient.

Reasons For:

- Quick realistic information on Procedures in the current System.
- Highlight aspects not detected in questionnaires/ interviews, meaning more detailed reports can be introduced.
- May be more reliable than interviews, can see what stakeholders actually do, instead of what they report to do.
- Help to understand positive and negative features of the Current System, which can be used when specifying the requirements of the New System.

Reasons Against:

- Time-consuming and expensive, an entire business or system cycle make be observed which could take significant time and administration costs.
- People act differently when they are being observed/ scrutinised, making some observations unreliable.
- Observations may be made by only one person, and thus may be biased.

Illustrating System Requirements (1.2.8)

Revised

PROTOTYPE

Simply, a Prototype is an early version or model of a System. It will typically:

- Display the minimum necessary features, used to test and gather feedback on a new concept or system from clients.
- Allows stakeholders (e.g. Client and Investors) to gain an idea of how the system would look/work, so they can prove feedback and suggest improvements. This will speed up the development process.
- It is often useful to produce more than one Prototype since this allows the manager to choose the Prototype which is most suitable, and thus provides more flexibility and improves the final system design.

Iteration (Iterative Design) (1.2.9 + 10)

Revised ☐

Where solutions, codes and prototypes are designed, developed, tested and evaluated in repeated cycles. With each iteration, additional features may be added until there is a fully functional software.

This involves end-user participation. Failure to involve end-users in the design process can lead to software not suitable for its intended use because of a lack of feedback. This has adverse effect on user productivity, efficiency and so on. Effective collaboration and communication between the client, developer and end-end-user is required.

Human Interaction with the System (1.2.12)

Revised ☐

Usability

Means making the computer systems easy to use, matching them more closely to user needs and requirements (examiners definition).

Ability to accomplish user goals. More usable means more efficient to use, easier to learn and acquire the functions of the system.

Accessibility

Ability of the system/device to meet needs of as many individuals as possible. Low accessibility means there are barriers to certain groups

(e.g. disabilities).

Usability Problems and Examples (1.2.13)

Revised ☐

Learnability

How easy is it to accomplish basic tests the first time users encounter the design?

(e.g. Learning features of different manufacturers, accidental touches on the touch screen, right-handed mouse, etc)

Efficiency

How quickly can users perform tasks?

(e.g. Need to locate product and details quickly on e-commerce sites)

Memorability	*When returning to design after a period of not using it, how easily can users establish proficiency?*
Error	*How many errors do users make, how severe, how easily can they recover?* (e.g. inaccurate/outdated street data, no verification/validation, time taken to reschedule)
Satisfaction	*How pleasant is it to use the design?* (e.g. visually appealing website, all other factors carried out)

Usability Problems for Different Devices

DEVICE	POTENTIAL ISSUE
GPS / Navigation System	Small Screen; Low-Quality Speakers; Poorly configured Antenna; Inaccurate Geographical Data; Outdated Street Data; Inefficient Routing Software;
Tablets	Accidental Touch/Input; Difficult to Learn Different Gestures; Poor Scaling and Zoom Control; Small Side Buttons; Poorly Written Instruction;
Game Consoles	Portable Consoles have Small Screens; Buttons may be too Small; Difficult to use Outdoors (Brightness); Short Battery Life;
PCs	Excessive use of Bright Screen leads to Eye Strain; Screen Reflection; Mouse is designed for Right-Handed People;
Digital Cameras	Incorrect calibration of Touch Screen Menu; Inadequate Flash; Buttons too Small; Software required to Share Files;
Mobile Phones	Very Small Keyboard; Not all users need Special Features; Battery Life;
MP3 Devices	Tiny Buttons; Insufficient Memory; Fragile; Lack of Screen in Micro Device Acceleration Sensors could increase Functionality;

Complexity/ Simplicity	(e.g. Unnecessary extra apps when the user just needs to call and use SMS, website clearly stating what company offers and clear navigation, unclear instructions)
Readability/ Comprehension	(e.g. Small screen/buttons, low-quality speakers, incomprehensible font and colours)
Other Factors	(e.g. Battery life, brightness outside, health problems, hardware components, camera without flash, insufficient memory, etc) - Small screen is difficult to see, especially in poor lighting. - Small keys make it hard to use and access functions. - Poor battery life inconvenient and needs constant charging.

Solving Accessibility/Usability Problems

ACCESSIBILITY ISSUE	POTENTIAL SOLUTION
Visual Impairment	Larger Screen; Braille Input Devices; Colour or Contrast Changes; Access to Hotkey Functionality; Text-to-Speech; Screen Magnification; Speech Recognition;
Hearing and Speech	Subtitles; Visual Effects; Voice Recognition;
Cognitive Problems	Word Processing to Correct Dyslexia; Special Software with strong Interaction; Use of Touch Screen and Bright Colours;
Mobility Issues	Eye Typer; Puff and Suck Switch; Foot Mouse; Special Speech Recognition; Word-Prediction Software;

Section 2

COMPUTER ORGANISATION

2.1 Computer Architecture

Central Processing Unit (CPU)

A hardware component of a computer system and can perform basic arithmetic, logical or input/output operations; essentially the 'brain' of the computer system. Also known as the 'Processor'.

The CPU contains the Control Unit (CU); Arithmetic Logic Unit (ALU); Memory Address Register (MAR); and Memory Data Register (MDR).

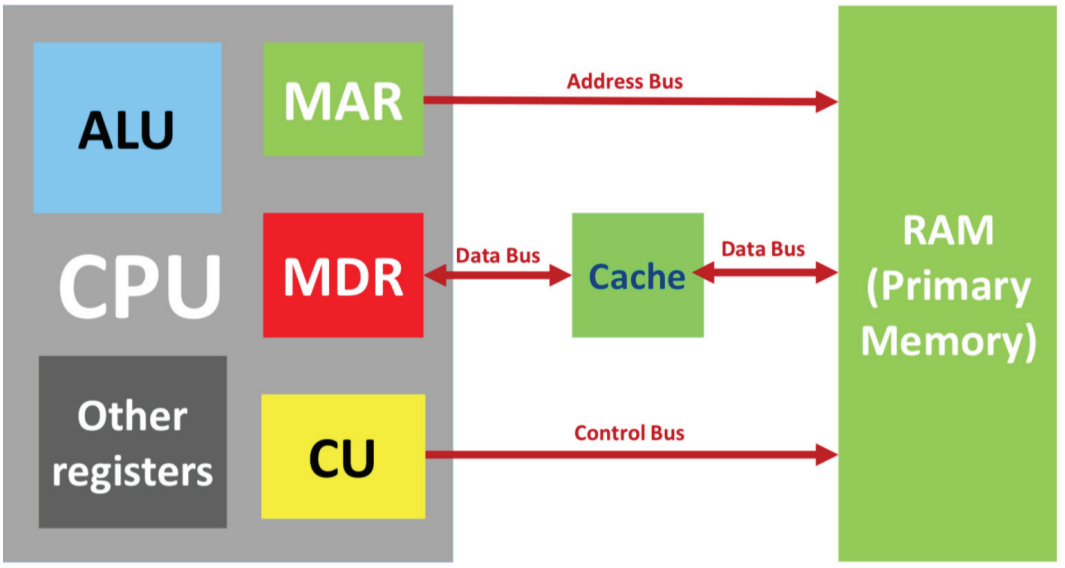
Control Unit (CU)

Fetches data or instruction from memory, decodes it into commands and controls transfer of among other units of the CPU (e.g. command to ALU for execution).

Part of the Processor which coordinates all activities within it.

Arithmetic Logic Unit (ALU)

Performs all basic Arithmetic and Logical and input/output Operations executes instructions.



Primary Memory (2.1.2)

Revised

Primary Memory is the only storage that is directly accessible by the CPU; at any point in time, it may hold both data and instructions that are currently running on the computer system. These instructions are stored in Machine Code (1s and 0s).

It consists of Two Types of Memory:

Random Access Memory (RAM)

Stores executing data and instructions of program currently being run. Data is stored in Unique Memory Locations; each of these Locations has an Address as well as Content.

RAM is a 'General Purpose' storage area, since data stored can be overwritten and instructions executed whenever necessary.

However, RAM is Volatile; therefore the contents of Memory are wiped whenever Power is lost.

Read Only Memory (ROM)

Stores permanent instructions and data of programs, used to boot and operate the computer (e.g. Basic Input Output System - BIOS).

Data can be Written-to not cannot be Overwritten; any ROM instructions are embedded. This makes ROM Non-Volatile — data and instructions remain even without Power.

By size, it is much smaller than RAM.

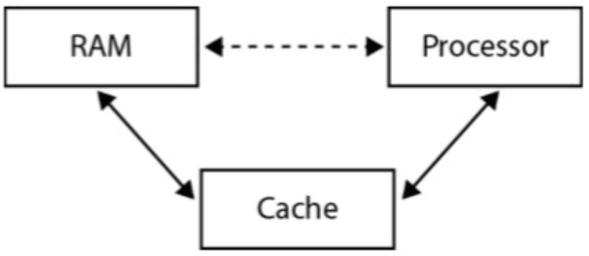
Cache Memory (2.1.3)

Revised

RAM can be further distinguished into Two Types:

1. Dynamic RAM (DRAM)
2. Static RAM (SRAM)

DRAM is used as the primary source of RAM in a Computer System. However, SRAM is faster; and a small section of SRAM, known as the Cache Memory is placed between the main RAM and the Processor:



As such, Cache Memory is Smaller and Faster SRAM that temporarily stores the most frequently used data and instructions so that the processor does not need to access the slower primary DRAM. 'Static' means that the memory doesn't need to be constantly refreshed.

When the processor (CPU) needs to read from the Primary Memory, it first checks if a copy of the data exists in the Cache; if so, it will read from the Cache instead of the DRAM. Cache Memory is located closer to the CPU providing faster access, and since the CPU accesses the slower DRAM less often, performance speeds up!

Therefore, Cache Memory is used to save time in accessing the RAM.

Control Unit (CU) Registers (2.1.1)

Revised

The Control Unit (CU) contains various registers. In General, a Register is a small storage location that can hold data (in multiples of 8 bits) — a 64-bit Computer would have 64 bit Registers.

The basic registers in the CU are the *Memory Address Register* (MAR) and the *Memory Data Register* (MDR):

Memory Address Register (MAR)

Stores the Memory *Address* of next data/instruction to be decoded and executed by the ALU. In order for the MAR to communicate with the Primary Memory (RAM), it connects through the *Memory Address Bus*.

Memory Data Register (MDR)

Stores the *Data* Itself taken most recently from the RAM, which is then used by the ALU. This allows the Processor (CPU) and Memory (RAM) to act independently, with the CPU not affected by differences in the speed of operation.

The connection between the RAM and the MDR is accomplished by the *Memory Data Bus*.

Machine Instruction Cycle (2.1.4)

Revised

Computer instructions are stored in the Primary Memory as a series of instructions in Machine Code (1s and 0s). These instructions/data have to be moved from the Primary Memory into the CPU in order for the Program to Operate.

The Basic Stages of Operation occur in a Cycle, known as the *Machine Instruction Cycle*:

1. **Fetching the Instruction**

   The CPU is responsible for knowing which instruction it needs to take from the Primary Memory. To achieve this, it sends the sends the instruction address through the *Memory Address Bus* (MAB) to the Primary Memory. The instruction inside the address is then copied into the *Memory Data Bus* (MDB) and sent to the Control Unit (CU).

2. **Decoding the Instruction**

   The instruction received by the CU is then decoded. Decoding an instruction allows the CPU to be aware of any additional data necessary for the execution of the instruction. Any additional data is appropriately fetched from the Primary Memory and delivered directly to the CPU through the same process as *Stage 1*.

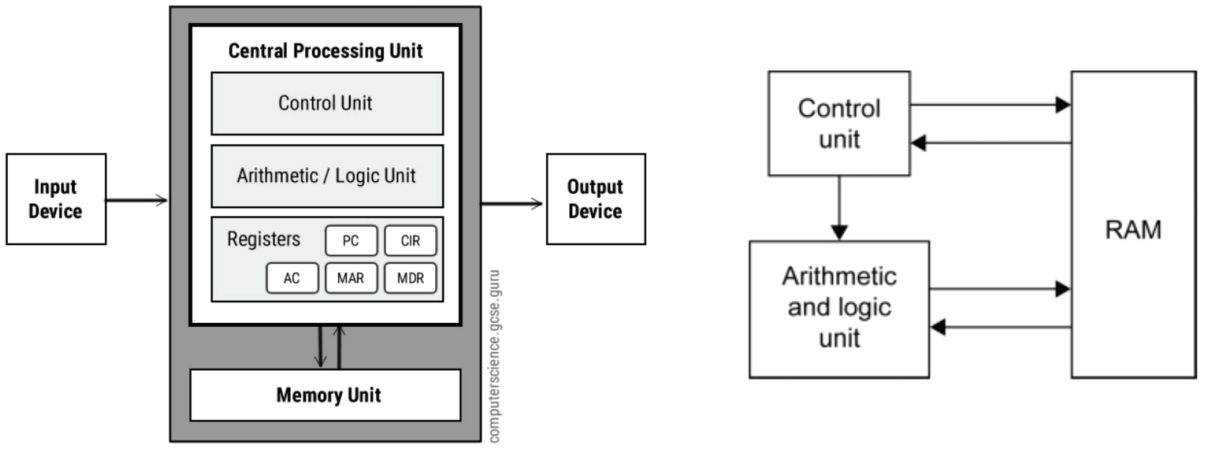
3. **Execute the Instruction**

   The CPU executes the instruction using the loaded data and calculates its result. Depending on the result, additional data may be needed; and *Stage 1s* processes occur again.

4. **Store the Instruction, then Check for Instruction**

   After executing the instruction and computing the result, the CPU stores the result in the Primary Memory. This is achieved through specifying the address the data will be stored using the MAB and sends this data through the MDB.

   The CPU then checks for the next instruction and repeats *Steps 1 to 4*



'Bus'

Simply, the Bus is a Set of Wires that connect to components in a computer system. Buses are used as physical connections to carry information to the CPU.

For example, the MDB transports data from/to the CPU, whereas the MAB states where the data is supposed to go/be.

Secondary Memory (2.1.5)

Revised

The Primary Memory is the only form of storage that is accessible by the CPU. This means that any data that is stored elsewhere need to be first copied onto the RAM, since ROM cannot be written-to but only read-from!

RAM is a volatile; so whenever power is lost, the memory contents of RAM are wiped. Also, RAM is a relatively fast memory form (read/write outputs and inputs, process tasks quickly), but has a limited memory capacity.

When RAM is full, Data and Instructions need to be saved elsewhere.

Secondary Memory, also known as Auxiliary Storage, is a relatively slow memory form that can be written-to (like RAM) but also is non-volatile (like ROM); therefore the contents are not wiped when power is lost. This makes Secondary Memory a form of Persistent Storage.

In comparison to Primary Memory, Secondary Memory has a relatively high storage capacity.

When a computer system starts up, the RAM is empty. Instructions and Data (such as the Operating System) need to be copied into the RAM in order for the computer system to run. In most computer systems, these instructions are copied from the Secondary Memory.

Consequently, any computer systems that need to hold persistent data cannot function with Primary Memory alone (e.g. Editing a saved file).

Examples include: *HDD* (Hard Disk Drive), *USB* (Universal Serial Bus), *SD* (Secure Digital), *Magnetic Tape* (Cassette, Floppy Disk, VHS).

Operating and Application Systems (2.1.6)

Revised

Operating System (OS) Set of software that controls computers hardware resources and provides services for computer programs.

The *OS* allocates (and de-allocates) specific sections of memory to each programs Process Module; this ensures that the memory assigned to one program is not overwritten. The *OS* uses Secondary Memory to allow more process to run simultaneously.

The *OS* is held in the either the Hard Disk; ROM or Solid-State Drive (SSD).

ROLES OF AN OPERATING SYSTEM

Peripheral Communication Peripheral Devices are all the hardware components of the computer system that exist outside the CPU. The OS is responsible for communicating directly with the hardware and providing an interface between hardware devices and applications.

Examples of Peripheral Devices:

- **Keyboard** (To type in data, or to type in a barcode if it doesn't scan).
- **Magnetic Card Reader** (used when a credit card is used).
- **Microphone** (to call the next customer, or to call the manager).
- **Monitor** (so the salesman can see the information/data on the screen).
- **Visual Display** (so the customer can read the information/data on the display).
- **Speakers** (for customers to hear information, for shop assistants to bring another item the customer may wish to buy).

Processor loading speed (efficiency) of OS functions need consideration, as does the graphics handling of the OS. The OS should be able handle input from appropriate devices (e.g. a developed video game application).

Memory Management An OS is responsible for all the memory that is available in a computer system, therefore it must Manage how Memory is used by applications, and ensures that one application does not interfere with memory that is being used by some other application.

The *OS* will allocate and de-allocate blocks of Memory to each Program being run, and keeps track of what data is in which location. This ensures that a Program has sufficient memory to run, and if there are too many processes for the RAM, the OS will use Secondary Memory (also known as *Virtual Memory*) to run programs simultaneously (e.g. Video Game applications use a lot of memory and require constant refreshing); this avoids overwriting or clashing of programs, and consequently optimises system performance and maximises memory usage.

Resource Allocation An application running on a computer system uses up Resources (including the amount of memory the application is occupying, or how much processing time is required). The OS is responsible for the efficient allocation of resources so that an application can run as effectively as possible.

Multiple applications may run on a computer system at once, and appear that they are performing tasks simultaneously. However, most computer systems only have a single *CPU* meaning that applications must share the CPU time in order to accomplish their process — this is known as computer *Multitasking*.

Networking The OS manages connections with networks of other computer systems to allow the sharing of Network Resources (e.g. digital files, printers).

Data Management The OS must have the ability to access data stored in Memory. Data is stored using files, and the OS must keep track of files and their location in order to make the best use of memory available, provide fast access times, and ensure that an application does not overwrite another applications files.

Security Provides measures such as password authentication, magnetic cards, access rights to prevent unauthorised access (e.g. Log files keep track of activity).

Software Application (2.1.7)

Revised

The main software applications that may be installed on a computer system include:

SOFTWARE APPLICATION	FEATURES
Word Processors (e.g. Microsoft Word, Google Docs)	Used for the production of documentation; Includes tools for composition, editing, formatting and printing of documents; Useful Autosave features mean that if power goes off, only data after the last (automatic) save is lost;
Spreadsheets (e.g. Microsoft Excel)	Data is represented as Cells, organised into Rows and Columns; can perform calculations through formulae to process data and present it in visual charts to analyse.
Database Management Systems DBMS (e.g. Microsoft Access)	Manages and provides interface for users to use database organised into records, and the user can create or modify these records, or query and extract data from them.
Web Browsers (e.g. Internet Explorer, Safari)	Used to access, retrieved and present content on the World-wide Web (WWW). Connects to web servers to request information.

Computer Aided Design CAD (e.g. Google Sketchup)	Often used in engineering, manufacturing and architecture to create, modify and analyse a design. Can convert info like shape, materials, dimensions, etc with changeable values. Supports the design/development of rapid prototyping in manufacturing a System solution; Saves time/costs associated to drawing and development; Photorealistic rendering/photo simulation in various industries (architecture/video games/ simulators/ etc) by using shading, radiosity, reflection, etc.
Graphic Processing (e.g. Adobe Photoshop)	Manipulate visual images with functions such as move, erase, crop, colour, etc.
Email (e.g. Microsoft Outlook)	Allows exchange of digital messages from a single author to one or more recipients

Common Features of Applications (2.1.8)

Revised

Graphical User Interface (GUI) Allow users to interact with software in different ways.

Command Line Interface (CLI) Where the user types in commands. Used in early software applications.

FEATURES OF AN OPERATING SYSTEM

Toolbar Buttons, icons, menus, etc.

Menu List of commands you can choose.

Dialogue Box Communicates information to the user and allows option choice.

These are common elements provided by the OS to improve usability as they're usually in the same place and used in the same way, while others may be provided by the software. Built in commands for inputting from touch screens; Predicted text so that typing a class name followed by a full stop will bring up a list of methods/attributes.

Automatically use colour to represent keywords/variables and improve readability.

GUIs provide ways that allows users to understand and interact with computer systems in a more natural and direct fashion. The common features of a GUI can be remembered using the acronym *WIMP* which stands for Windows, Icons, Menus and Pointers.

Section 3

NETWORKS

3.1 Network Fundamentals

Computer Network A computer system or set of systems which uses communication equipment in order to connect computers and share their resources and data.

Node Device on a Network

Server A Computer program, Software or Host Computer that fulfils requests from client programs and serves as a Central Repository of data and programs which are shared by these clients.

Client Computer Hardware or Software Device that accesses a service made available by a server, achieved by sending a request for service. (e.g. email software client requesting email server software to fetch new emails).

Unlike a Server, the Client does not share any of its resources, but requests content.

Hub Connection point for all devices on network, connected through Ethernet cables. Data from one device goes direct to the Hub, which share this to all devices.

The intended recipient accepts this data, all other devices ignore it.

Although transmission to every port ensures that data reaches its destination, the effect of this creates a lot of traffic on the network, which slows down transmission.

Switch	Also the connection point for multiple devices on a single network; however a Switch can identify which device is connected to which port, so can send Data to the Target Device directly. Because of this, Networks connected using a Switch have faster data transmission than those connected using a Hub.
Router	Used to join Multiple Networks and is used as an intermediary between these so data may be exchanged efficiently between devices (e.g. Connecting a home Network to the Internet).

Type of Networks (3.1.1)

Revised

Local Area Network (LAN)	Network devices are connected within a Limited Geographical Area (e.g. a room, home, school, office building, etc). Usually client-server mode of operation, with Shared Data and Resources (e.g. Peripheral Devices).
Peripheral Devices	Examples include Printers, Scanners, External Hard Drives which provide increased functionality.
Virtual Local Area Network (VLAN)	Different sections of a LAN may not want to access all the available shared data (e.g. Different departments in a School), as such a LAN can be split into separate VLANs, the new network is known as a Logical Separate Network. Each Logical Separate Network cannot access the other shared resources (e.g. Data, Peripherals) without being granted special access.

Wide Area Network (WAN/GAN)	Connects different computer systems or LANs from different geographical areas, can span over a city, country or the world (e.g. internet, different sites of organisations connected). A WAN would require Security; using login, passwords, security questions and encryption.
Storage Area Network (SAN)	Type of LAN created so that large storage devices can be accessible from servers in a convenient and easy way, and is able to handle large data transfers.
Personal Area Network (PAN)	Network covering individuals Workspace, essentially a LAN which only supports one person, instead of a group of people (e.g. Peripheral Devices connected through USB/Bluetooth).
Peer-To-Peer Network (P2P)	Network with no centralised server. All nodes are client and server at the same time, consuming and supplying resources from and to other peers. This makes resources more widely Available and supports File Sharing for Collaborative Work.

WIRELESS LOCAL AREA NETWORK (WLAN)

Like a LAN, a WLAN Connects nodes in a Limited Geographical area but achieves this without the use of cabling and wires.

This requires technology such as a wireless Router or Modem; Access points; Switch; Wireless Repeaters and so on.

Network Reliability depends on the strength of wireless signal and distance from router, and on the type of Network Topology and shape of surroundings. Furthermore, multiple connection points on the Network may reduce transmission speeds.

Advantages:

- Possibility of user collaboration and make access easier due to wireless transmission. Quick access with mobile devices; Anywhere in the scope of the WLAN and therefore do not need to be present at a workstation.
- Allows users to bring their own device which has a more familiar interface, could lead to greater efficiencies and fewer usability issues.

  Could lead to insecure devices, but could be resolved with clear company policy regarding use, Virtual Testing 'Sandbox' could protect against Viruses, MAC address authentication and testing, other security features in order to address.
- No extra equipment is needed for expansion after the initial set-up; which will save time and money.
- Reduces wiring therefore increases the safety for employees.
- User-mobility and Economical access points.

Disadvantages:

- Poorer Security as devices from outside can access the Network and Intercept Transmissions, since transmission goes 'through the air'.

  Could be resolved by strong encryption/protocols, e.g. WPA-2 or use of trusted MAC addresses, and regular changes of the router password.
- Open to Misuse; as an administrator cannot directly monitor a specific user or device.
- Data transfer will Decrease (compared with a wired LAN) because the number of computers using the network increases and because the WLAN has a lower bandwidth than a wired LAN.
- Intermittent Connectivity due to physical barriers; this may result in Lower Transfer Speed, Weakening of Signal and may reduce Operations and Efficiency.
- Wireless signal could be weak in some placed; Leading to frustrated/ineffective users.

Internet

A Global WAN connected through an Internet Service Provider (ISP) in exchange for a monthly fee (Broadband cost).

The Internet is decentralised by design, any independent Computer able to access the internet and share resources becomes a Server of its own.

Ethical Issues include: Some information presented online may be deliberately incorrect and not subjected to validation and scrutiny; Further issues such as plagiarism and Intellectual Property.

Extranet

Computer network that utilises the Internet. Works like an External extension to a LAN. Part of a network that uses internet protocols to allow Limited access by specific users to a LAN or WAN.

(e.g. Business wants to share some data/ information with clients or partners but not all, so extends part of the network, creating an Extranet, available for access but with security/privacy measures).

Ethernet

Ethernet is more reliable as the strength of the signal is independent from the distance from the router (in comparison to Wireless Network transmission). There is also no issue with the network topology type and nature of the surroundings, so long as the user is connected. Connection does depend on the condition of the cables; any loose or broken cable connections will reduce or fail transmission.

The Speed of Transmission across a Network may vary due to different parts of the Network using different Media, Network congestion, Data Packets taking different routes; The Receiver may be busy or simply the Physical size of the Network.

Virtual Private Network (VPN) (3.1.5)

Revised

A VPN allows clients from distant locations to connect, that otherwise wouldn't be able to connect with LAN (too far through cables) or WAN (too far for signal to be picked up). Tunnelling allows the clients device to appear to be a node of the internal Network; thus affording the client full access to the Network resources. The Network protocols (e.g. IPSec or TSL) are still secure despite passing over an outside Network.

Transmission is always encrypted and provides a secure connection; established between sender and receiver (both sender and receiver are authenticated before transmission); therefore any unauthorised access will not be able to understand the data.

Data is hidden through the use of hidden IP Addresses, Encrypted connections, multiple Exit Nodes, Tunnelling (sending a Packet within a Packet and encrypting it).

Technology Required to Provide VPN (3.1.4)

Revised

- SSL 3.0 (Secure Socket Layer 3);

  TLS (with encryption - Transport Layer Security);

  IPSec (Internet Protocol Security) to encrypt and authenticate traffic over virtual tunnels.

  Requires special Client Software, whereas SSL and TLS are supported by all Web Browsers.

- Tunnelling protocols; Allows the data to be encapsulated (hidden) whilst travelling across the internet. Encryption protocols (IPSec) if hacked will not be decipherable. The use of Gateways allows the person to connect with the central Server.

- Hardware for pubic networks like the internet through tunnelling, which allows the network to send data via other networks connections as if connected to a LAN.

- Hardware/Software requirements like internet access, VPN software, routers.

Advantages of a VPN:

- Information can be accessed in remote places.
- Lower Cost; No need for long-distance leased lines.
- Enhanced Security; data intercepted is undecipherable and security properties of each tunnel are agreed by the administrators of the two Endpoints of the Tunnel.
- Nobody outside the VPN should be able to affect the security properties of the VPN.
- Multiple Exit Nodes; makes it hard to distinguish where the Data was Generated and thus More Secure.
- Can change Working Patterns; allowing Employees to work from home and reduce travelling time, beneficial for Workplace Efficiency and reduces Environmental Costs.

Disadvantages of a VPN:

- Needs professional with detailed understanding of security issues and configuration to ensure sufficient security and protection.
- Reliability of VPN is not directly under the organisations control, but under the ISP.
- Not all VPN products are compatible across different vendors.

VPN VS EXTRANET	
Access and Transmission are always Encrypted	Limited level of Encryption
Users have Access to Everything	Only have access to Enabled and Specific Services
Authenticate the Sender before establishing a Tunnel	

Importance of Standards in Construction of Networks (3.1.2)

Revised

Common Rules and Standards are crucial to the formation of a Network, otherwise Systems may not be able to connect and communicate due to Incompatibility.

Standards play an important role in the construction of Networks; and can be thought of a common 'Language' that enables compatibility.

Network Communication Layers (3.1.3)

Revised

An application goes through different layers to send data between systems.

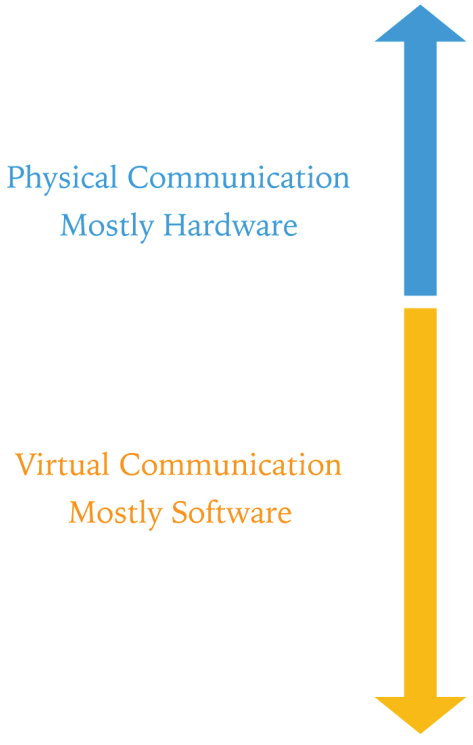
If one software application is trying to send Data to another software application, it must be encoded into a format which is understandable by both software applications; to achieve this data is broken into Packets (which contain the data and the address of its destination).

Each layer has its own protocols; and the most widely used Networking Standard is the OSI Model and works as a reference which describes and explains each stage of the network communication process.

Open Systems Interconnection (OSI)	Model This model define layers of network interaction.

The OSI is a standardised system/model for network connection and consists of 7 Layers, each dealing with specific parts of Network Communication (for example the physical layer which defines the physical connection).

1. **Physical** *(Cabling system components).*
2. **Data Link** *(Network Interface Card).*
3. **Network** *(Routing).*
4. **Transport** (*Transmission, Error detection).*
5. **Session** *(Retransmission of data if not received by device).*
6. **Presentation** *(Encryption and decryption of messages for security).*
7. **Application** *(The end-end-receiver application, e.g. email).*



3.2 Data Transmission

Unit of data for transmission with a format, it is part of a message made into a single package. Contains address and data.

It contains a set amount of data; it contains a fixed structure (or identify elements of the structure other than data); it contains data that is to be sent via a communications channel; it also contains specific details for transmission (e.g. address of sender and receiver, error codes, etc).

Data Switching (3.1.11)

Revised

Network communication method: Original file is divided into packets before transmission, each of these packets may follow a different path to the destination address.

- Packet Switching involves Splitting Data into Packets and sending these from source to destination through different routes, using Network Switches and Routers

- Network Switches and Routers are intermediate devices which determine how best to transfer the packet on the path to its Destination (rather than flowing over a singe wire).

- In order to achieve this, there are Rules and Standards used to compile and transmit each packet in a Standard Format. Packets must be constructed in exactly the same way so that the Receiver knows automatically how to decode its contents, and does not require further instruction for decoding the Packets.

- These will have Protocols that contain the Receiver Address, so the Packet knows where it needs to be sent and Packet Numbers which identify which Sequence the Packets are sent in.

NEED TO KNOW:

- **Address;**

  Have to be in standard format so that each switch/ routing station recognises the address.

- **Address of Sender;**

  Identifies the sending computer, so that any packets not received can be re-requested.

- **Address of Receiver;**

  Identifies intended recipient so it can be forwarded on correctly.

- **Protocol;**

  Must be identified so that the correct rules are followed.

- **Size of packet/fields;**

  All packets/fields must have the same size so that data can be reassembled.

- **Sequence Number;**

  So that packets can be reassembled in correct order.

- **Transmission Codes;**

  To show whether the data packet is transmitted or re-transmitted.

- **Control Bits;**

  To Maintain the integrity of the data by ensuring that the data received is the same as the data sent.

- **Error-Checking Code;**

  When an error is detected, an algorithm either corrects the error or requests that the packet is resent.

Protocols (3.1.7)

Revised

Sets of rules for transmission; to facilitate a process being carried out correctly.

In the case of Data Transfer, protocols are rules that ensure data is transferred correctly between systems. A protocol recognised as the standard for a certain type of transfer is called a 'Standard Protocol'.

Protocols ensure the presence of an identified Sender and Receiver and the agreed Method of Communication. It also provides rules about Format; Data Compression; Error Checking/Validation and the Recovery and Resending of Data.

WHY PROTOCOLS ARE NECESSARY:

- **Data Integrity**

  Ensures data is not changed or corrupted during transmission.

- **Flow Control**

  Network infrastructures have limited memory and bandwidth, the Transport Layer (4) is responsible for using Protocols to manage situations where an overload of resources occurs.

  (e.g. Transport Layer could request a Sending Application to slow down its data flow rate to manage Network Congestion).

- **Deadlock Prevention**

  Prevents situation where two or more Network competing actions are each waiting for the other to finish, and thus neither ever does.

- **Congestion Management**

  Prevents requests on network resources from exceeding capacity.

- **Error Checking**

  Ensures the validity of data

EXAMPLES OF INTERNET PROTOCOLS:

- **Hypertext Transfer Protocol (HTTPS)**

  Creates a secure transmission of data from Client to Server.

- **Secure Socket Layer (SSL) and Transport Layer Security (TLS)**

  Encryption protocols used on the Internet. Allows for things like secure payment.

- **Internet Protocol Security (IPSec)**

  Encrypt and authenticate traffic to ensure secure transfer over VPN tunnel.

- **Dynamic Host Configuration Protocol (DHCP)**

  Allows the server to automatically assign IP address to client device.

Data Compression (3.1.9)

Revised ☐

Data Compression reduces the size of files to be transmitted over a Network; which will take up less Bandwidth and reduce overall Transmission Time. The recipient will have a program that can decompress the contents of the File.

Lossy Compression — Permanently deletes certain information, only part of the original data is displayed when compressed.

This is an acceptable solution for formats such as JPEG, GIF, MP3 as the difference isn't easily noticed.

Lossless Compression — Only eliminates data which has 'statistical redundancy', all original data is still available when decompressed.

This is typically used for Word files, which are small and easily decompressed.

TRANSMISSION MEDIA

Wireless — Compared to metal cabling (Ethernet) and Fibre Optics, wireless is the least reliable and slowest; but is the cheapest form of transmission.

(e.g. Microwave, Radio Signals, Satellites, WiFi).

Metal Conductor — Faster, more reliable and expensive than Wireless but still cheaper than Fibre Optic cabling.

(e.g. Copper cable, UTP Cable, Coaxial (TV) Cable).

Fibre Optics — Fine optical Fibres carrying beams of Light as signals. Fastest, most Reliable and Secure but also the most Expensive transmission option.

FACTORS THAT AFFECT SPEED OF TRANSMISSION

- Bandwidth (theoretical speed of the Network).
- Data transfer Rate of Storage Devices (interference, traffic, number of connected devices, errors, type of Files, malicious software).
- PC System Performance (CPU Speed; RAM)

3.4 Network Security

Encryption Altering a message into undecipherable form to those unauthorised. Only a recipient with the correct Key can decode the message and read it.

Symmetric-Key Encryption (3.1.15)

Revised

The same Single or 'Secret' Key is used for Encryption and Decryption; each device uses this to Encrypt a packet before it is sent over an Untrusted Network.

Reasons For:

- Faster than using a Public Key
- Uses less computer resources

Reasons Against:

- Keys must be shared before they are used.
- Risk or Key becoming known by unauthorised individuals, another one must be used.

Public Key Encryption (3.1.16)

Revised

Also known as 'Asymmetric-Key Encryption'. A Public Key for Encryption and a Private Key for decryption are both mathematically linked. SSL and TSL Protocols use this form of Encryption.

Reasons For:

- Both sender and recipient don't need to share Key beforehand to communicate.

Reasons Against:

- Messages take longer to encrypt and decrypt.
- Authenticity of public key needs to be verified.

'MEDIA ACCESS CONTROL' ADDRESS (MAC)

- The MAC Address identifies a specific device through its unique 12 Characters (via Network Card/Controller) which is checked against a list of approved addresses — if not on this Whitelist then Access is Denied.

- This prevents Unauthorised Access very difficult, unless the NIC has been cloned; but generally provides an Additional Layer of Security since data sent to a specific MAC Address can only be accessed on the specified device.

Reasons For:

- Extra Security

Reasons Against:

- Danger of allowed list of MAC addresses being discovered.
- Difficult to manage the list as it grows.

User ID — Uses password to access networking device and/or another password to access the web interface used to set up a Wireless Router or Access Point.

Reasons For:

- Easy to use.
- Prevents unauthorised access.

With web interface password, a person cannot access web-based utility page of router/modem/ access point unless they have the password.

Reasons Against:

- Entering password each time may be time consuming and inconvenient.
- Weak passwords are easy to crack and gain access.

Firewall

Either as; Software installed on each device, or Router Hardware firewall that protects from Hackers accessing devices through wireless connection.

Filters incoming traffic and can block some messages coming through, as well as control/limit users own access to the internet. Extent of firewall use depends on an Organisation's Policy

(e.g. One organisation ay not only allow communication between users and outside the network via email, but restrict website access also)

Reasons For:

- Software firewall monitors traffic between device and network and prevents unauthorised access.
- Router hardware firewall prevents unauthorised persons getting access to the network.

Reasons Against:

- May slow down the device.
- Issues about censorship with software firewall, depending on the organisation.

Router hardware firewall prevents unauthorised persons getting access to the network.

METHODS TO INCREASE NETWORK SECURITY

- Giving each user appropriate Login Details and Strong Passwords.
- Different Access Rights for Students, Teachers, Admins, etc.
- Encryption of all Passwords and Files.
- Use of the latest WiFi Protocol (WPA2) Security.
- Require MAC Address and Authentication.

Section 4

APPLICATION OF COMPUTER SCIENCE

4.1 Binary Representation

Binary Digit (Bit)

Basic unit of information, represented as 1 or 0.

Byte: 8 bits Kilobyte: 1000 bytes Megabyte: 1000 kilobytes

Minimal unit of storage that can be set to 1, or 0.

Decimal or 'Denary'

Positional system that uses 10 digits to represent a number. Also known as 'Base 10'.

Binary has only two states: 0 or 1, which is why it is known as 'Base 2' - however Decimal has ten possible states: 0 to 9, which is why it is known as 'Base 10'.

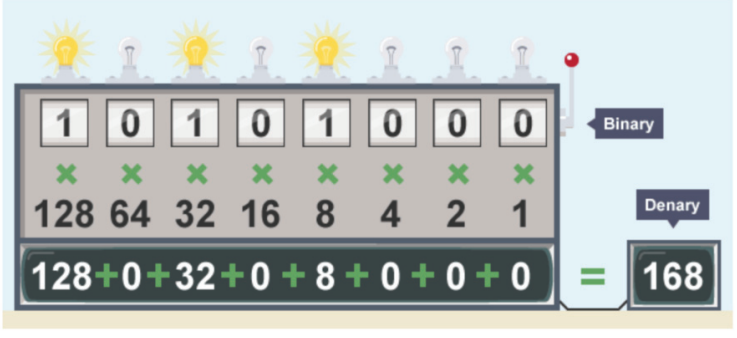
Thousands 1000s (10^3)	Hundreds 100s (10^2)	Tens 10s (10^1)	Ones 1s (10^0)
6	4	3	2

Or think of it as:

$(6 \times 1000) + (4 \times 100) + (3 \times 10) + (2 \times 1) = 6432$

Each base has its own placeholders. For example, in Decimal 6432 can be thought of as 6 1000s, 4 100s, 3 10s and 2 1s.

In Binary the same is true:



- Nibble - 4 bits (half a byte)
- Byte - 8 bits
- Kilobyte (KB) - 1000 bytes
- Megabyte (MB) - 1000 kilobytes
- Gigabyte (GB) - 1000 megabytes
- Terabyte (TB) - 1000 gigabytes

The values fall into the binary pattern (1, 2, … 128) like Connect Four… putting it into the highest value then putting the remainder into the next… and so on.

Here, 10101000 becomes 168 in Decimal.

Binary Representation (2.1.10)

Revised

In computers, an 'on' state, when electricity is flowing, represents true. The 'off' state, no electricity flowing, represents false. This can be represented with 0 and 1.

8 BITS

0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0
128	64	32	16	8	4	2	1	0

= 256 states / values

Integers 256 values can be represented by 8-bit numbers. When representing negative numbers, the most significant bit (MSB, leftmost bit) is either 0 (positive) or 1 (negative) so only 7 bits are left for the number.

Characters American Standard Code for Information Interchange (ASCII) character encoding scheme represents text. Each character (lower or uppercase latin letters, numbers, punctuation and control characters) is assigned a value from 0 to 127. Unicode can also have symbols and other languages.

Strings (e.g. UTF-8 Unicode is used - on average 40 bits per word is required).

Colours Each pixel has a colour value, often represented by a Hexadecimal RGB 6-digit number (3 values), every two digits show how much red, green or blue there is in the colour. Can also be represented in binary; more bits means more colours. Resolution is the width * height in pixels.

A colour will be split into three components (Accept RGB as an example); Each component will be assigned a certain number of bytes.

BASE PLACEHOLDERS

Denary *("Base 10")*	.. 1000 (10^3)	100 (10^2)	10 (10^1)	1 (10^0)
Binary *("Base 2")*	.. 8 (2^3)	4 (2^2)	2 (2^1)	1 (2^0)
Hex *("Base 16")*	.. 4096 (16^3)	256 (16^2)	16 (16^1)	1 (16^0)

HEXADECIMAL

This was created to simplify the way Binary is represented.
It is 'Base 16' due to having digits:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Each Hex digit represents 4 Binary digits:

DENARY	BINARY	HEXADECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

For example; D4 is 11010100

Hexadecimal is only really used for Human comprehension.
It is always converted back into binary for mach interpretation.

Hexadecimal can be used to represe colours using the format #RRGGBB with 256 possible shades for each red, green, or blue - and over 16 million colours in total (256 ^3).

16^3	16^2	16^1	16^0
4096	256 X 1	16 X B	1 X 5

Hexadecimal representation of 1B5 - decimal value = (1*256)+(11*16)+(5*1) = 437

Algorithm: keep dividing by 16 and keep remainder, then put them together (write letters for d over 9). Put the first value into the rightmost box.

Example:

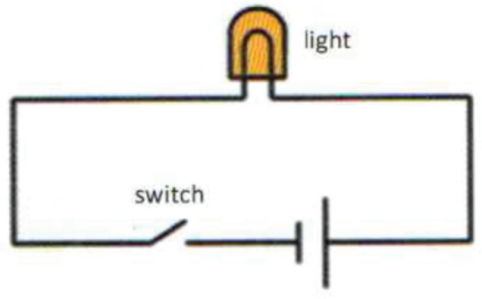
Using placeholders, we can convert any denary number into Binary or Hex

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		
	128	64	32	16	8	4	2	1		
Binary	0	1	1	0	1	1	0	1	=	109

	16^7	16^6	16^5	16^4	16^3	16^2	16^1	16^0		
	-	-	-	-	4096	256	16	1		
Hex	-	-	-	-	-	-	6	D	=	109

BOOLEAN LOGIC (2.1.11)

Revised



Transistors are electrically controlled switches, made up of two electrodes and one control wire:

When you apply electricity to the control wire, it lets current flow through one electrode through the transistor, to the other electrode.

The control wire can be considered to be the input, and the wire from the bottom electrode as the output.When the input is true, the output is true. When the input is false, the output is false: this is the most basic Boolean Logic.

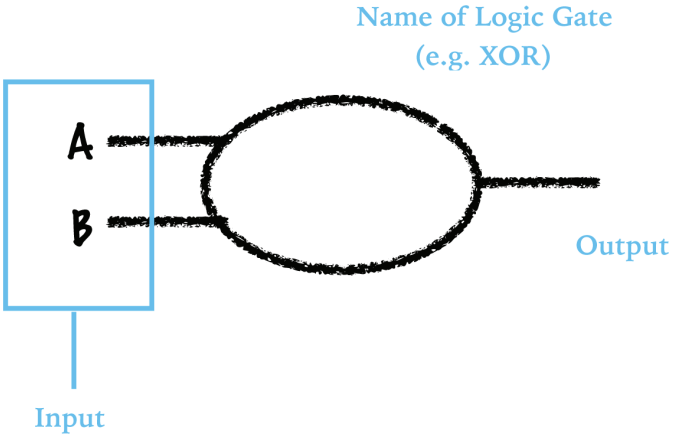
AND	$\wedge$	Both A and B must be true for the output to be true
OR	$\vee$	Either A or B or Both may be true for the output to be true
XOR (Exclusive Or)	$\veebar$	Either A or B may be true for the output to be true, but NOT BOTH
NAND (Negation And)	$\neg\wedge$	Negation of output for AND
NOR (Negation Or)	$\neg\vee$	Negation of output for OR

DRAWING LOGIC GATES

When asked to draw a Logic Gate diagram, shape specifics do not matter.

- Inputs are always leftmost.
- 'OR' is always the last (rightmost) gate, when drawing multiple Logic Gates at once.

Boolean Logic is given as 1 (true) or 0 (false).



Concurrent Processing (4.1.14-16)

Revised

This is when one or more processes are using the same resources at the same time (e.g. Building a house, Production lines, Division of labour). Concurrent Processing may be evaluated according to if it has saved time/resources.

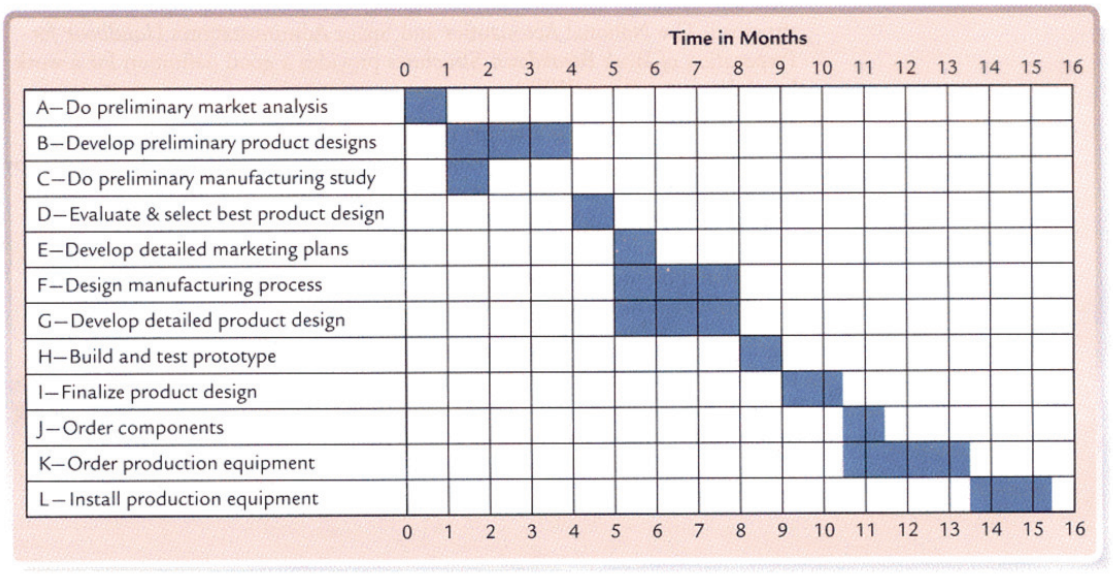
In a Concurrent Processing system, one job uses several processors to execute sets of instructions in parallel. The advantages of Concurrent Processing are:

- Increases computation speed.
- Increases complexity of programming language and hardware (machine communication).
- Reduces complexity of working with array operations within loops, conducting parallel searches in databases, sorting files etc.

On the GANTT Chart, any tasks carried out at the same time have been carried out Concurrently. In contrast, any tasks which are not carried out Concurrently have been carried out in Sequence, and are thus examples of **Sequential Programming**.

GANTT CHARTS

Here, the horizontal axis represents the span (time) of the project, and is broken into increments such as days, weeks or months. The vertical axis represents the tasks that make up the project.



Reason For:

- Efficient organisation; helps establish time-frame.
- Highly visual, easy to stick to.

Reason Against:

- Potentially over-complex.
- Needs to be constantly updated if project requirements change.
- Doesn't show whole picture; details of task.

Batch Processing:

Large amount of input happens over time, then whole set of input is processed in one go.

e.g. Monthly Mobile phone Bills (all inputs/access is added up and charged at the end of the month)

Online Processing:

Input processed almost immediately.

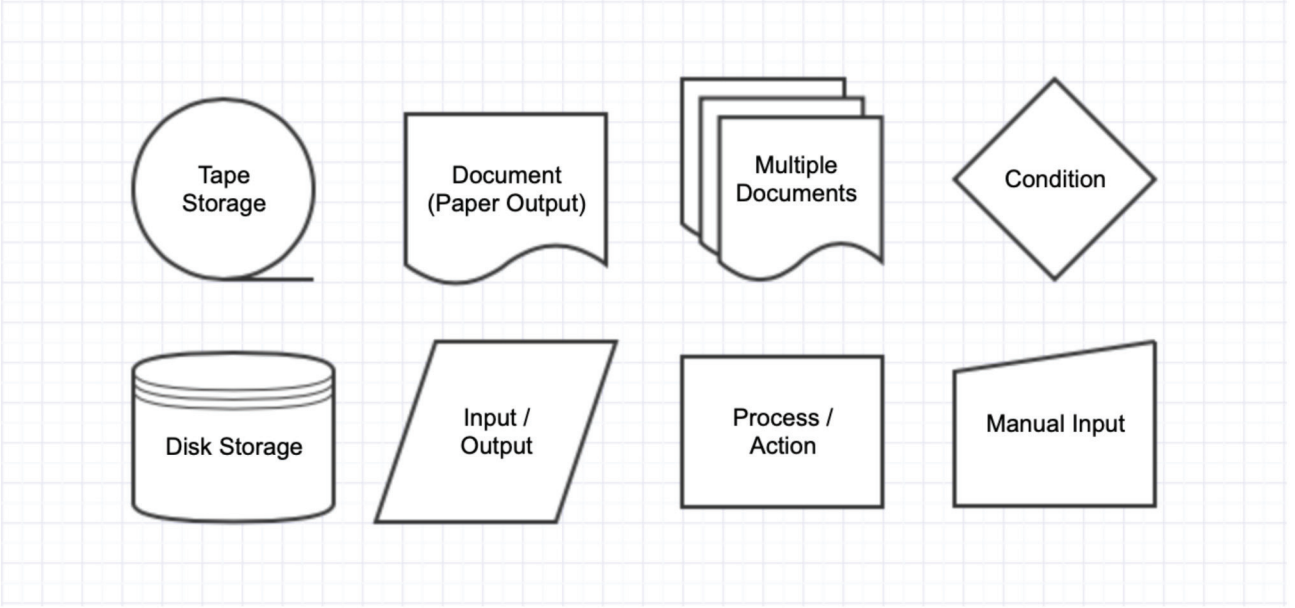
e.g. Flight seat booking system; as soon as you book the seat is yours.

Real-Time Processing:

Input processed immediately and continuously; generally without any user input, of which comes from system sensors.

e.g. Autopilot on Aircraft; large volume data harvested from multiple sensors continuously, system reacts in real time.

FLOWCHART SYMBOLS



TYPES OF FLOWCHARTS

Algorithm Flowchart

Each process represented by a flowchart symbol, stated in detail.

System Flowchart

Most basic, general overview of the system itself, with no detail.

Abstraction (4.1.17)

Revised

Abstraction is the process of taking away or removing characteristics from something in order to reduce it to a set of essential characteristics. This enables one to concentrate on the essential aspects of the problem, whilst ignoring any distracting details. It is a simple way of dealing with complexity.

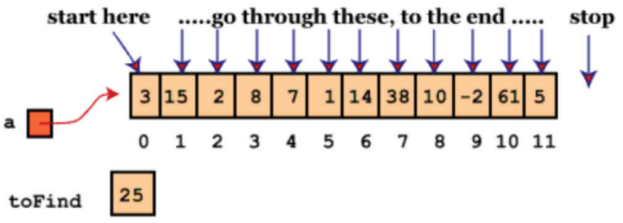
Standard Algorithms on Linear Arrays (4.2.1)

Revised

SEQUENTIAL SEARCH (A.K.A "LINEAR TO FIND")

Also known as a 'Linear Search'; this is an algorithm used to find an item in a list.

- It starts with the first element and compares each element to the one it's looking for, until it is found.

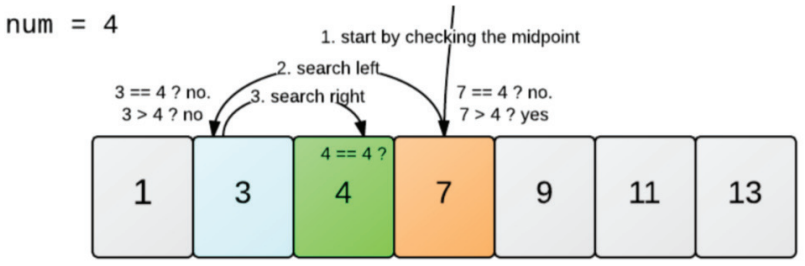



```
NAMES = "Bob","Betty","Kim","Lucy","Dave"

output "These names start with D"

loop while NAMES.hasNext()
    NAME = NAMES.getNext()
    if firstLetter(NAME) = "D" then
      output NAME
    end if
end loop
```

BINARY SEARCH (A.K.A "SORTED AND SPLIT")

Also known as a 'Half-Interval Search'; this is a search algorithm that finds the position of a target value within a sorted array. Crucially, it only applies to sorted arrays, where there are no duplicate values; or where duplicates do not matter.

- It works by comparing the target value to the middle element of the array;
- If they are unequal, the lower of upper half of the array is eliminated depending on the result and he search is repeated in the remaining sub-array until it is successful.



BUBBLE SORT (A.K.A "SWAP TO SORT")

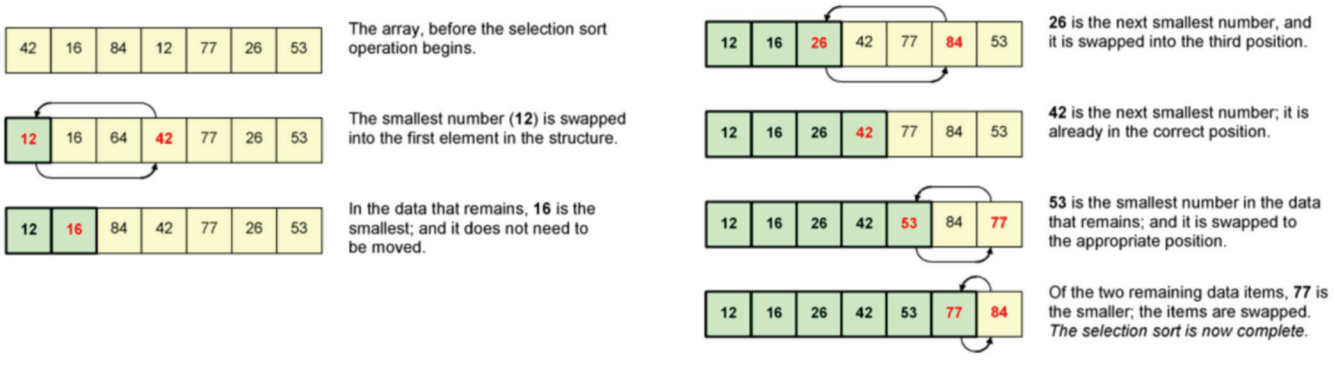
Known as 'Bubble Sort' because of way smaller elements bubble to the top of the list. Although the algorithm is simple in nature, it is often too slow and impractical for most problems.

- This a sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order.
- It will repeatedly pass through the list until there are no swaps needed, which indicates that the list is sorted.

SELECTION SORT (A.K.A "SUBLIST TO SORT")

- Algorithm divides the input list into two parts: the sublist of items already sorted, which is built up from left to right; initially the sorted sublist is empty and the unsorted sublist is the entire input list.

- The algorithm proceeds by finding the smallest (or largest, depending on sorting order) element in the unsorted sublist and swapping it with the leftmost unsorted element; then moving the sublist boundaries one element to the right.



Higher Level Language and Machine Code (4.3.5)

Revised

Compiler

When a high-level language is translated into lower-level language (done in a batch).

Translation into machine code for final execution.

Interpreter

When a high-level language is translated into an intermediate code which will be immediately executed by the CPU (done line by line).

Warns syntax error, shows outputs for tested processes.

For example, the Java Virtual Machine allows the same Java code to be installed on various different types of hardware; this means the programmer does not need to write many different versions of the program for each different device type.

Use of Programming Languages (4.3.6)

Revised

Variable — a Storage Location for Data in a program. They are a way of naming a memory location for later usage (to put a value into). Each variable has a name and a data type that is determined at its creation and cannot be changed.

Constant — an Identifier with an associated value which cannot be altered by the program during execution (the value is constant). This is the opposite of a *variable*.

Operator — a Set of characters which represents an Action. Types include:

- **Boolean**:
  `(AND/OR && ||)`
- **Arithmetic Operators:**
  `(+/- div mod)`
- **Assignment Operator:**
  `(=)`
- **Relational Operators**:
  `(> == != .equals() )`

Discuss the Need for Sub-Programmes and Collections (4.3.12)

Revised

Modular Programming is a beneficial approach to programming problems; as it allows the program to be easily organised (for the coder and team members); makes future maintenance easier and is efficient in its use of reusable code.

By breaking it down into modules, each compartment can be easily designed and tested. This also allows for 'distributed development' where multiple programmers can work in parallel.

Standard Operations of Collections (4.2.2)

Revised

Collections are unordered lists of unknown length or size. In Java, these collections are known as 'LinkedLists'; which are useful when you don't know how many items you'll be needing/using (compared to an array of fixed length). It is also an efficient use of RAM Memory and can be of any data type. It has operators (collection methods) such as:

- `addItem(data)` adds data item to the collection
- `resetNext()` starts at the beginning
- `hasNext()` tells whether there is another item in the list
- `getNext()` retrieves a data item from the collection
- `isEmpty()` check whether collection is empty

These are predefined Sub-programs, known as 'Collection Methods'

OBJECTS AS A PROGRAMMING CONCEPT

Revised

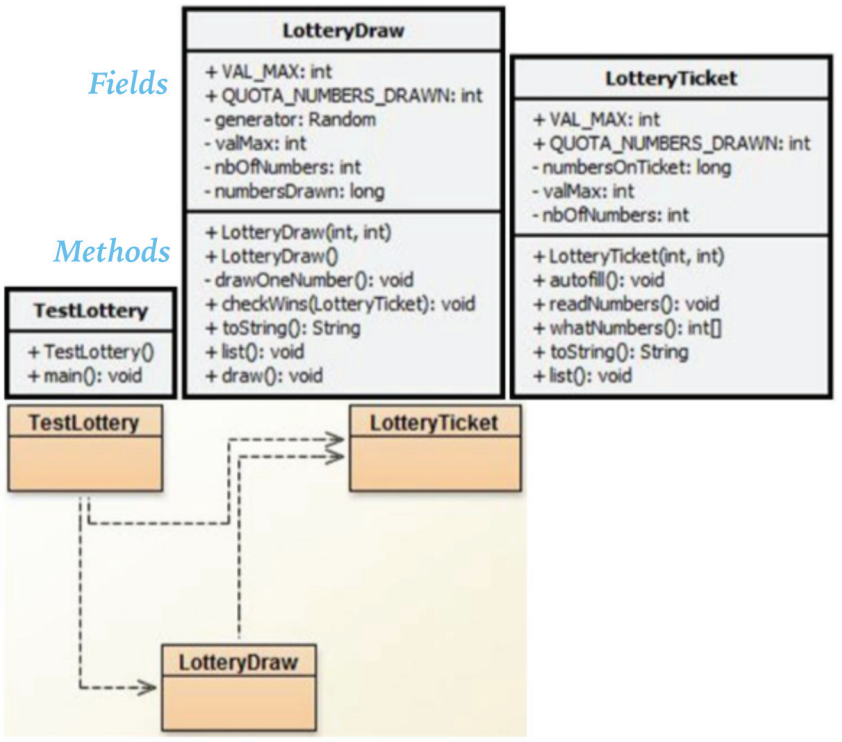
CLASSES AND OBJECTS (D1.1/2)

Revised

Object an Object is a representation of a real world entity (e.g. Book, Car, Student)

Class a template for creating these objects. A Class identifies what data/attributes need to be stored for these objects (this data are the Fields) and the procedures/functions/behaviours needed to access this data (known as the Methods).

UML DIAGRAMS



Here, the grey boxes are the UML Diagrams for the Classes.

The top half show the Fields and the bottom half show the Methods.

The notation (+) shows the variables are Public and (-) Private.

The Yellow Boxes show the Relationship between the Classes.

The constructor method of the Class has the same name as the Class (here it is TestLottery).

When a program is executed, the term Object has a different meaning. Here, an Object is created by a constructor method of the class; and this created Object is known as the **instance of the class**.

Here, the Class is a template for a lottery draw but would create a new lottery Object each draw.

Each draw would have its own set of winning numbers. This is how Objects are created, each has the same structure but its attributes (Fields) would be different (e.g different numbers).

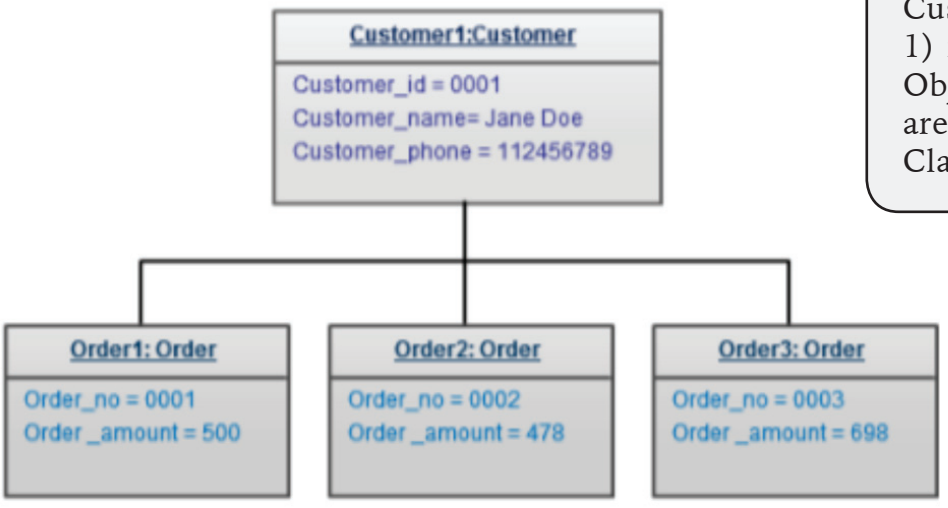
UML Diagrams (D1.3/4)

Revised

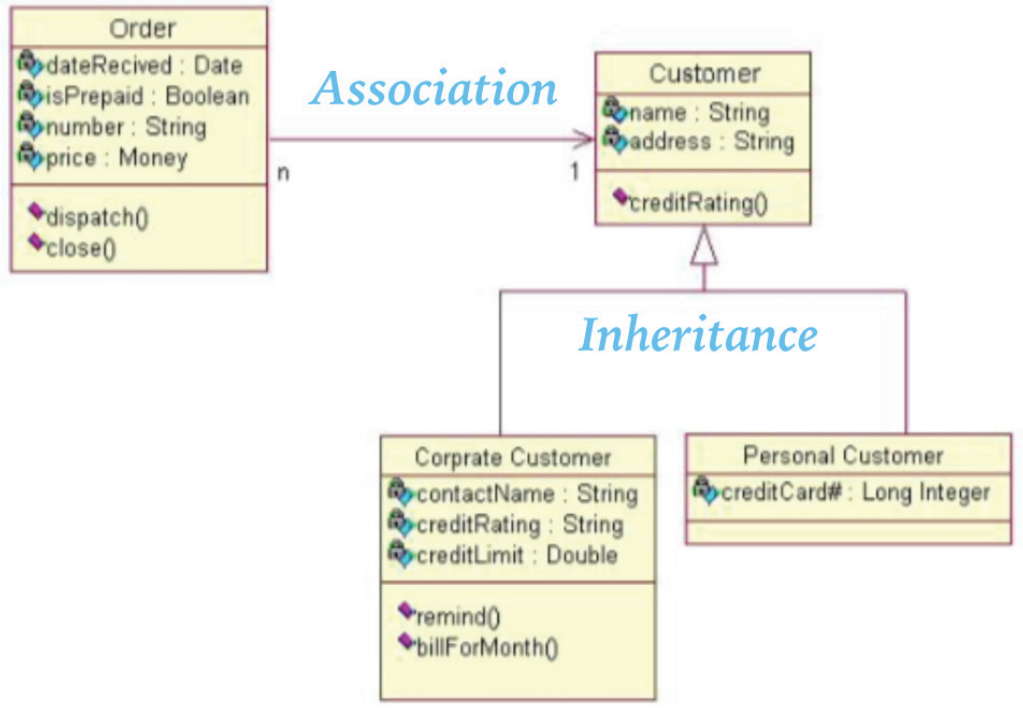
UML Diagrams are used to visualise computer thinking and program design. At IB, UML Diagrams are distinguished into two types:

OBJECT DIAGRAM

This is a diagram for Objects inside the Class. Here, the Customer Object (Customer 1) is associated with 3 other Objects (Order 1-3), which are instances of the Customer Class.



CLASS DIAGRAM



A Class diagram shows all the separate Classes inside a Program.

Each Class lists its Fields; for example Order has the Fields dateRecieved of type Date, isPrepaid of type Boolean, and so on.

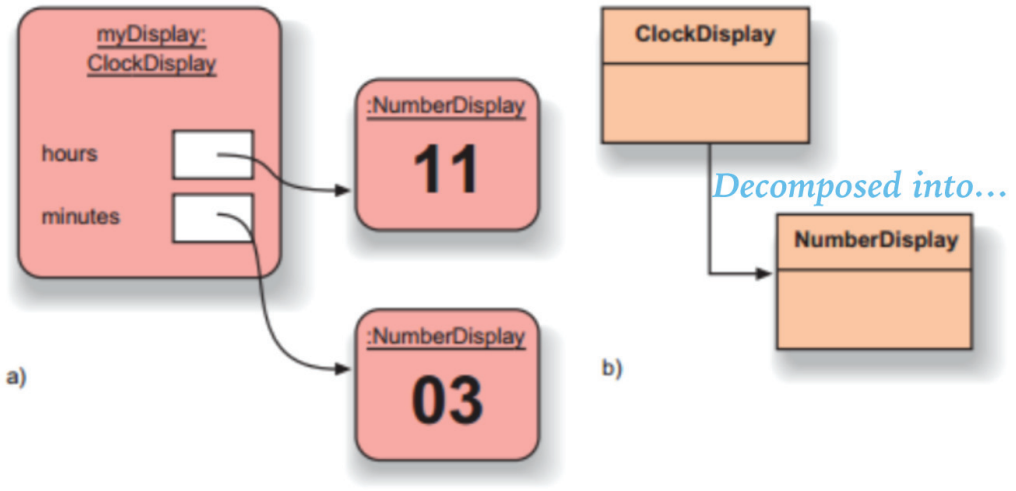
The Order Class has Methods dispatch() and close(), which show you have access to them. The name of the Method typically indicates what it does.

The Customer Class has Fields of Name and Address both of type String and a method called creditRating(). The Classes Corporate Customer and Personal Customer are types of the Customer Class (shown by the Relationship Arrows) and so **inherit the Fields and Methods** of the Customer Class.

These Classes Corporate Customer and Personal Customer may have their own specific Fields of which they do not share, for example creditLimit of type Double.

Decomposition into Related Objects (D1.5)

Revised



An example to demonstrate Decomposition into Related Objects can be seen with the 'Digital Clock'.

The clock is made of two 2-digit number displays. So Class numberDisplay was created to hold these two 2-digit numbers alongside Class numberDisplay.

For example, a Calendar is made up of days which can be grouped into weeks and grouped into months; this shows that you have decomposed the Calendar in days, weeks and months.

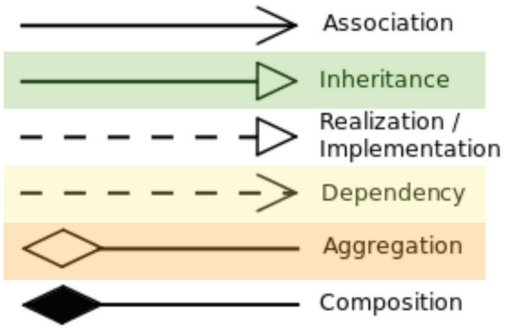
Describe the Relationship between Objects (D1.6)

Revised

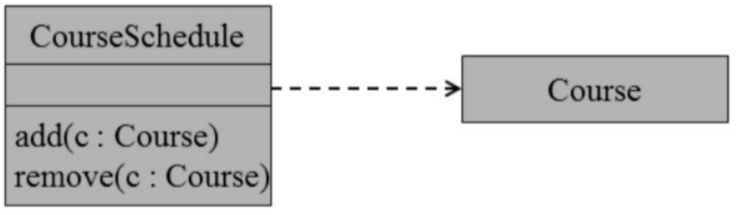
Objects can have three types of relationships:

- Dependency (x uses y)
- Aggregation (x has a y)
- Inheritance (x is a y)

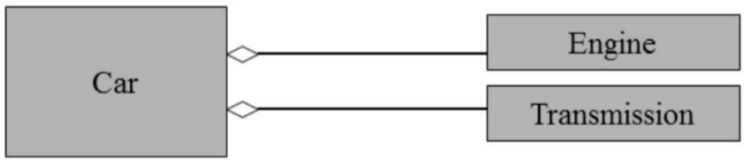
UML NOTATION



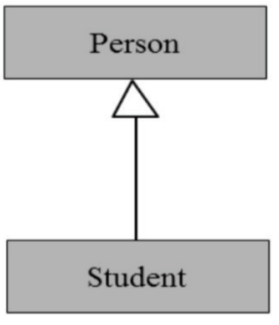
Dependency:



Aggregation:



Inheritance:



Here Class courseSchedule depends on Class course because the add() and remove() Methods both use the Class course.

Here Class Car has (includes) Fields of type Class Engine and Class Transmission (a Class can be a type).

A Class can inherit Fields and Methods from a super-Class.

Here the Class Student inherits the Fields and Methods from Class Person, but Class Student may have additional Fields and Methods itself.

Outline the Need to Reduce Dependencies between Objects (D1.7)

Revised ☐

Whenever Class A uses Class B, then A depends on B. Therefore, Class A is the 'Dependent' and Class B is the 'Dependency'.

Two Classes which use one another are 'Coupled'. The degrees of the relationship between Coupled Classes is continuous, not discrete.

Dependencies are bad because they decrease reuse. Reuse is found in code where parts are reiterated or have a replicated structure; this is helpful in code readability; code quality, etc.

Changes in one class will affect other (dependent) classes… which may increase errors in code.

Describe How Data can be Pass to/from Actions as Parameters (D1.10)

Revised ☐

Actions are Methods; and some need data it does not have direct access to. Because of this, specific data can be sent into the Method as an argument and accepted as a parameter.

Advantages of Libraries of Objects (D2.7)

Revised ☐

Libraries are made available for parts of code that are likely to be reused. These are imported into Java programs, and provide increased functionality. They work like pre-made methods.

import java.lang.Math.*;

Describe the Disadvantages of Object-Oriented Programming (D2.8)

Revised ☐

- **Size**

  OOP are much larger than other programs; and in the infancy of computer technology when storage capacity was poor, this made OOP unfavourable. Today, this is not an issue.

- **Effort**

  OOP require a great level of planning before any code is written.

- **Speed**

  OOP typically run slower, due to their size.

Discuss the Use of Programming Teams (D2.9/10)

Revised ☐

The modular nature of OOP means that objects can bee assigned to a different coder to program, as long as any shared Methods are known. This allows further efficiencies in being able to finish coding the program faster and allowing users to specialise to their strengths. Additionally, each module can be tested and debugged separately.

However, this could cause communication/coordination issues; diffusion of responsibility; conflict.

Program Development

Definition of Terms (D3.1-3)

Revised ☐

Class	Defines the Fields and Methods for a group of similar objects. An object is an instance of a Class.
Identifier	A broad term for the name of a Variable or Method.
Primitive (Data Type)	byte, int, long, double, char, boolean, etc
Instance Variable	The Fields in a Class. So when an object is created of that class type, it has its own version of the Fields.
Parameter Variable	Information sent into a method from the calling program. `pubic int add(int x, int y)`
Local Variable	A Variable which exists only as long as the code to which it belongs exists. So if declared in a Method which is no longer being run; the variable no longer exists/ it is overwritten.

Method A program routine or behaviour contained within an Object designed to perform a particular task on the fields within an object.

Accessor A Method that allows access to the fields within an object.

```
getName()
```

Mutator A Method that allows the values of the files in an object to be changed.

```
setName()
```

Constuctor The Method that correctly initialises and sets up an object when it is created. This method has the same name as the class it belongs to, and has no return data type.

```
public person()
```

Signature The Method name and its parameter types.

```
doSomething(String y)
```

Return Value The value returned by a method to the calling program.

Private Variables, Methods, Fields and Constructors that can only be accessed within the declared class itself. Classes cannot be private. An object is able to encapsulate itself and hide its data from outside access.

Protected Variables, Methods, Fields and Constructors declared protected in the super-Class can only be accessed by the sub-Classes.

Public Variables, Methods, Fields and Constructors declared public can be accessed from any other Class.

Extends This is used by a sub-Class to say that it inherits the fields and methods from a super-Class.

Static In Java, a static is a member of a class which isn't associated with the instance of a class. It belongs to the Class itself. As such, you can access the static member without first creating a class instance.

Modern Programming and Ethical Considerations (D3.9/10)

Revised

When organisations interact on an international basis, there may be issues of language differences.

- **Common character sets**

  Character sets used among many platforms and languages, like UNICODE. Unlike ASCII, UNICODE has a much larger number of characters so has letters and symbols from different alphabets/ languages.

- **Platform independent languages:**

  High level programming languages like Java enable code to run on many different platforms.

ETHICAL/MORAL OBLIGATIONS OF PROGRAMMERS

- **Adequate testing:**

  To prevent possibilities of commercial or other damage.

- **Plagiarism**

  Acknowledge the work of other programmers to avoid being accused of plagiarism.

- **Open source movement**

  Increasing support for open-source software, where you are given access to the code for free, allowing you to make changes/customisations to meet exact needs.

4.3 Java Object Oriented Programming

FOUNDATIONS OF OBJECT ORIENTED PROGRAMMING:

Object — A software bundle of related state (data) and behaviour (procedures). These procedures and functions (attributes) are known as Methods. It is also the 'Instance of a Class'.

In the real-world everything has state and behaviour, for example Dog has state (name, colour, breed) and behaviour (wagging, eating, running). What states can this object have, and what behaviours can it perform? In Java (object oriented programming) everything is an Object.

Advantages of objects:

- Objects can be easily passed around the program, and so are modular in nature.
- Allows for re-use for objects already created.
- Debugging ease; if a bicycle is broken you replace the bolt, not the whole bike. Similarly, you can replace and diagnose specific objects.

Members — Objects have their own variables, constants and methods associated to it. These are broadly known as 'Members' or 'Features' of an Object.

Class — A category of objects (template) for which objects are created. Models the state (data) and behaviour (methods) of a real-world object.

In the real world, there are thousands of Bicycles (Objects) for which all were built from the same set of blueprints and thus contain the same components. In OOP terms; your bicycle is the instance of the class of objects known as 'Bicycles'.

A Class is the blueprint for which individual bicycles (Objects) are created.

It can also hold Identifiers which are shared by, and accessible to all the Objects within the Class (static members) and can handle methods which deal with the entire Class rather than a specific object.

Inheritance The capability of defining a new class of objects that inherit state and behaviours from a parent/super-class.

New data and methods can be added to the class, whilst the data and methods of the parent class are available for objects in the new class.

Different kinds of objects often have certain amounts in common with one another. Mountain bikes, Road bikes and Tandem bikes all share characteristics of bicycles (speed, cadence, etc) — yet each has additional features that makes each different. Bicycle is the parent/super-Class of MountainBike and so on.

public class B extends A

This means that class B will have all the members (variables, constants and methods) in class A. 'extends' shows that class B inherits from class A; alternatively it can be said that class B is a sub Class of the parent Class, class A.

Advantages of objects:

- No need to re-write methods when sub-Classes can extend a parent Class.
- Any changes in the parent Class are inherited by the sub-Class automatically.

Encapsulation Mechanisms that allow each object to have its own data and methods (states and behaviours). Encapsulation is affected by having all method calls handled by objets that recognise the method.

Data and Methods are limited to the object in which they are defined. Private variables can be only accessed outside of its class by specific Accessor and Mutator methods.

```
void add(Data dat, Key k)
```

Called using: `t.add(dat, k)` *Receiver…* *…Message*

Where the method call is known as the 'message' and t is the 'receiver' of the message. The benefit of this approach is that there can be many methods named 'add' with different objects implementing them in different ways. This allows programmers to reuse names of methods.

Advantages of objects:

- Can restrict the way data/methods are altered or called — ensures variables are not accidentally changed by another part of the program, whilst still maintaining their functionality.
- Hide the way data is stored.
- Since relevant data/methods are linked makes it clearer to diagnose any run problems.

ACCESS MODIFIERS:

In Java, the keyword public or private indicates the level of accessibility. When private is used, the member will not be inherited by the sub-Classes.

Polymorphism The capability of having Methods with the same names exhibit different behaviours depending on the 'receiver'. Can send the same message to two different objects and they can respond (return) in different ways.

Commonly achieved through 'Overloading' which allows two objects to have the same name but different functionality. Values can be passed through the parameters (braces) of the method call and return a specific output — this changes the Method Signature which allows for Polymorphism.

Advantages of objects:

- External programs can use subclass methods without knowing any details.
- Reusability of same Method names with differing functionality.

JAVA CONCEPTS AND STATEMENTS

Variables Objects store their states in 'Fields' — these are also known as 'Identifiers' and include variables (value can change) and constants (value cannot change).

Instance Variables Also known as 'Non-static Fields' (declared without static keyword). Instance variables are unique to each instance of the class (Object).

`currentSpeed()` of one bicycle is independent of the `currentSpeed()` of another bicycle object.

Class Variables Also known as 'Static Fields' and tells the Compiler there is exactly one copy of this variable only.

`static int numGears = 6` could be declared since the same number of gears will apply to all instances.

Local Variables Similar to how an object stores its state in Fields, a Method will store its temporary state in Local Variables. Any variables stored within the braces '( )' of a Method are Local Variables.

Parameters Set the conditions for a program operation. Can be sent across Methods within the Class as an 'argument'.

```
System.out.
println("Hello
World!")
```

Here, "Hello World!" is the condition for which it will be printed.

Primitive Data Types: (D1.9)

Revised

In Java, all variables must first be declared before they can be used, by stating the variables type and name.

`int gear = 1` tells the program that a Field named 'gear' exists, holds numerical data and has a value of 1.

A Primitive Data Type is predefined by Java and is stated using a keyword. Those within the scope of SL IB Computer Science are:

`byte` an 8-bit group of binary digits operated on as a single unit.

`int` an Integer, which is any whole negative or positive number.

`float` numbers with fractional value (a rational number with finite decimal expansion).

`double` numbers with a decimal place

`boolean` only two possible values; true or false.

`char` a single character (can take range of Unicode characters)

`String` text values written inside quotations.

Operators Variables can be evaluated using Operators, which include:

Symbol	Definition	Examples	
=	is equal to	X = 4, X = K	If X = 4
>	is greater than	X > 4	if X > 4 then
>=	is greater than or equal to	X >= 6	loop while X >= 6
<	is less than	VALUE[Y] < 7	loop until VALUE[Y] < 7
<=	is less than or equal to	VALUE[] <=12	if VALUE[Y] <= 12 then
≠	not equal to	X ≠ 4, X ≠ K	
AND	logical AND	A AND B	if X < 7 AND Y > 2 then
OR	logical OR	A OR B	if X < 7 OR Y > 2 then
NOT	logical NOT	NOT A	if NOT X = 7 then
mod	modulo	15 mod 7 = 1	if VALUE[Y] mod 7 = 0 then
div	integer part of quotient	15 div 7 = 2	if VALUE[Y] div 7 = 2 then

Relational Operators `(==, >, =>, <, <=, !=)`

Arithmetic Operators `(+, –, * , / , % , =)`

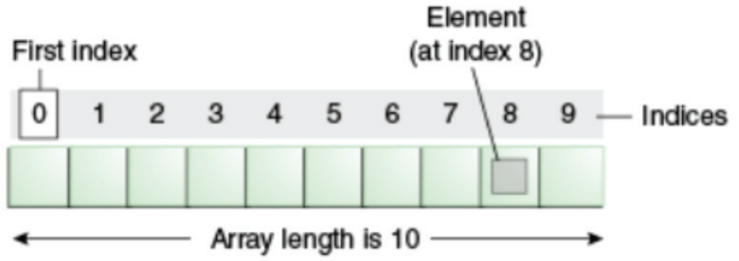
Unary Operators `( ++, - - , !)`

Conditional Operators `( && , | | )`

Arrays

An Array is a container object that holds a fixed number of values of a single type.

The length of the array is established when it is created, therefore its length is fixed.



An array of 10 elements.

Each item in the Array is known as the 'Element', and each Element is accessed by its numerical 'Index'. Numbering begins with 0, so the element: 9 is accessed at index: 8

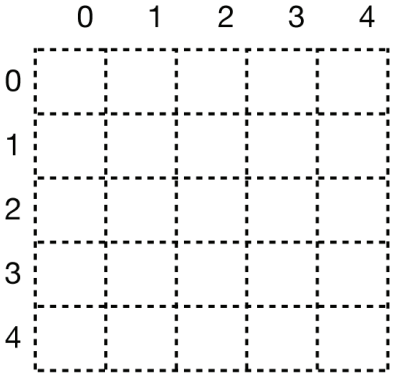
The Array type is expressed as: `type[]`

Where type is any data type and creates elements of this type.

`String[] stringArray` would create an Array of String values.

Multidimensional Arrays can be created using two or more sets of brackets:

`String[][] stringMultiArray` would look like:



ARRAY STATEMENT EXAMPLES

Create array w/ random values:

```
double[] arrayName = new double[10];
for (int j = 0; j<10; j++){
      arrayName[j] = Math.random();
}
```

Printing array values:

```
for(int = 0; j<10; j++){
      System.out.println(arrayName[j]);
```

Find average of array values:

```
double sum = 0.0;
      for (int j = 0; j<10; j++){
            sum = sum + arrayName[j];
      }
double average = sum/10;
```

Finding a min or max value:

```
for (int j = 0; j<10; j++){
      if(arrayName[j]>num)then{
            num = arrayName[j];
      }
```

EXPRESSIONS:

A Construct made up of variables, operators and method invocations which evaluate to a value.

Expressions are the core components of Statements; and these statements may be grouped into 'Statement Blocks'.

STATEMENTS:

Statements are like sentences, it is a complete unit of execution; each line in a program is a statement.

Methods are defined by the statements within them, since these create their behaviour.

Statements specify a sequence of actions which should be performed when the method is invoked.

STATEMENT BLOCKS:

Any sequence of statements can be grouped together to function as a single statement when expressed within braces — { }.

A statement block can be written inside another statement, this is known as 'nesting'. They are used to define methods.

```
double payRate = 15.75; {
      System.out.println(payRate);
      double newPayRate;
      newPayRate = 12.75;
      System.out.println(newPayRate)
}
```

CONTROL FLOW STATEMENTS:

Statements are executed sequentially, on their own. However, Java has control statements which break the sequential flow of code, and allow repetitive execution of statements and conditional execution of statements.

DECISION-MAKING STATEMENTS:

if-then	Only executes code if test (written in parameters) evaluates to true. If false, skips to end of if statement.
if-then-else	Same as if-then but goes to a second condition if test evaluates to false rather than skipping the statements.
case-switch-break/default	A Switch Statement can have any number of possible execution paths for int, char and byte data types.

LOOPING STATEMENTS:

while	Evaluates an expression, which must return a boolean value. The while statement continues testing the expression and executing the statement block beneath it until it evaluates to false.
for	A compact way to iterate over a range of values expressed in the parameters:

```
for(int i=1; i<11; i++){
     System.out.println("Count is: " + i);
          }
     }
```

BRANCHING STATEMENTS:

break	Terminates a loop when a value is found or statement block complete (expressed at the end of the statement block).
return	Exits from the current method and returns to where the method was originally invoked. By specifying a value with return, the exit can be achieved whilst sending the value back (e.g return num1;)

OBJECT CREATION:

A Class contains Constructor Methods that are invoked to create objects from the Class template.

Constructor Methods always share the same declaration name as the Class.

To create a new Bicycle object called myBike, the Constructor Method is called by the new operator. Creating an Object is the same as 'instantiating a Class'.

```
public Bicycle(int startGear, int startSpeed)
                        {

. . .

Bicycle myBike = new Bicycle(2, 0);
```

Therefore myBike is an instance of the class of Bicycles and has attributes startGear of 2 and startSpeed of 0 both of type int.

OBJECT REFERENCE:

In order to refer to the Fields within an object outside of the current class, an 'object reference' expression must be used with the dot operator:

Fields

```
motorBike.startGear;
```

You can use 'object reference' to invoke (call) an Object's Method. Additionally, any arguments within the parentheses can be provided:

Methods

```
myBike.setName(nameBike);
```

Parameters are the list of variables in a method declaration (also typically separated by commas).

Arguments are the actual values passed in when the method is invoked.

When an argument, stated inside the parameters, is passed through the Method; the internal processes of the Method use the argument temporarily in order to perform a calculation.