Unit 4 Digital Solutions: Preparation for the external exam

Teacher resource

August 2020
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>4</td>
</tr>
<tr>
<td>Assessment objectives</td>
<td>5</td>
</tr>
<tr>
<td>Example Exam Paper Resources</td>
<td>6</td>
</tr>
<tr>
<td>Concept One Focus: Encryption and authentication strategies</td>
<td>7</td>
</tr>
<tr>
<td>Objectives</td>
<td>7</td>
</tr>
<tr>
<td>Resources</td>
<td>7</td>
</tr>
<tr>
<td>Activities</td>
<td>9</td>
</tr>
<tr>
<td>Practice Exam Questions</td>
<td>10</td>
</tr>
<tr>
<td>Concept Two Focus: Useability Principles &amp; Elements and Principles of Visual Communication</td>
<td>13</td>
</tr>
<tr>
<td>Objectives</td>
<td>13</td>
</tr>
<tr>
<td>Resources</td>
<td>13</td>
</tr>
<tr>
<td>Activities</td>
<td>13</td>
</tr>
<tr>
<td>Practice Exam Questions</td>
<td>14</td>
</tr>
<tr>
<td>Concept Three Focus: Security of Data</td>
<td>16</td>
</tr>
<tr>
<td>Objectives</td>
<td>16</td>
</tr>
<tr>
<td>Resources</td>
<td>16</td>
</tr>
<tr>
<td>Activities</td>
<td>16</td>
</tr>
<tr>
<td>Concept Four Focus: Data Flow Diagrams</td>
<td>17</td>
</tr>
<tr>
<td>Objectives</td>
<td>17</td>
</tr>
<tr>
<td>Resources</td>
<td>17</td>
</tr>
<tr>
<td>Activities</td>
<td>18</td>
</tr>
<tr>
<td>Practice Exam Questions</td>
<td>18</td>
</tr>
<tr>
<td>Concept Five Focus: Algorithms</td>
<td>21</td>
</tr>
<tr>
<td>Objectives</td>
<td>21</td>
</tr>
<tr>
<td>Resources</td>
<td>21</td>
</tr>
<tr>
<td>Activities</td>
<td>22</td>
</tr>
<tr>
<td>Practice Exam Questions</td>
<td>25</td>
</tr>
<tr>
<td>Concept Six Focus: Pseudocode</td>
<td>28</td>
</tr>
<tr>
<td>Objectives</td>
<td>28</td>
</tr>
<tr>
<td>Resources</td>
<td>28</td>
</tr>
<tr>
<td>Activities</td>
<td>28</td>
</tr>
<tr>
<td>Practice Exam Questions</td>
<td>29</td>
</tr>
</tbody>
</table>
Introduction

This resource has been developed to assist teachers facilitating Digital Solutions Unit 4 to prepare students for external assessment. It has been prepared with the help of Leigh Ferguson, Leader of Learning in Technologies, Stuartholme School; Maggie Golawska-Loye, Brisbane Girls Grammar School and resources from the Brisbane Digital Solutions Hub.

This resource has also been designed to be a ‘living document’ and we encourage teacher input. If you would like to add to the document please contact accelerators@qut.edu.au. We will endeavour to update the resource biannually.

Disclaimer

QUT makes no representations about the suitability of this information for any purpose. It is provided "as is" without express or implied warranty.

QUT disclaims all warranties with regard to this information, including all implied warranties of merchantability and fitness, in no event shall QUT be liable or any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this information. This information may include technical inaccuracies or typographical errors. We may make improvements and/or changes to our information at any time.

Links to external websites

This resource contains links to other sites that are external to QUT. We take care when linking to external websites, but we have no direct control over the content of the linked web sites, or to the changes that may occur to the content on those sites. It is the responsibility of the user to make their own decisions about the accuracy, currency, reliability and correctness of information contained in linked external websites.

Copyright

© Queensland University of Technology

Unless otherwise noted or marked, copyright material available in this website is "all rights reserved". Apart from noted or marked exceptions or fair dealing as permitted under the copyright law of your country, and as necessary for the use of this resource, no part of this resource, nor the information contained in it, may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, reprographic or otherwise, without the prior permission of the Queensland University of Technology. To notify us of a copyright problem refer to the Copyright information at https://www.qut.edu.au/additional/copyright

To request copyright permission, email accelerators@qut.edu.au
Assessment objectives

The external assessment technique is used to determine student achievement in the following objectives:

1. Recognise and describe programming elements, components of exchange systems, privacy principles and data exchange processes
2. Symbolise and explain programming ideas, data specifications, data exchange processes, and data flow within and between systems
3. Analyse problems and information related to a digital problem
4. Not assessed
5. Synthesise information and ideas to determine possible low-fidelity components to secure data exchange solutions
6. Not assessed
7. Evaluate impacts, components and solutions against criteria to make refinements and justified recommendations
8. Not assessed

Unit 4 has three topics:

- **Topic One**: Digital Methods for Exchanging Data
- **Topic Two**: Complex Digital Data Exchange Problems and Solution Requirements
- **Topic Three**: Prototype Digital Data Exchanges

Please note:

Not all subject content from Unit 4 will be assessed for the External Assessment (EA) item. The Unit 4 EA objectives have been aligned with the subject matter from Units 1, 2 and 3.
Example Exam Paper Resources

Example Papers from NSW

HSC Information Processing and Technology
Data structures, DFD, data transmission protocols, SQL, security, normalisation, impacts


HSC Software Design and Development
Algorithms, data structures, DFD, encryption, design principles, impacts


Example Exam papers from VIC

Computing Software Development
Data structures, XML, data transmission, algorithms


Informatics


The BBC Bitesize site revision pages

Computer Science

➤ https://www.bbc.co.uk/bitesize/subjects/z34k7ty

Digital Technologies

➤ https://www.bbc.co.uk/bitesize/subjects/z9qy6yc

American AP Computer Science
Revision for the exam

➤ https://www.khanacademy.org/computing/ap-computer-science-principles
Concept One Focus: Encryption and authentication strategies

Objectives

Topic One

- Recognise and describe
  - Encryption and authentication strategies appropriate for securing data transmissions and their differences
  - Features of symmetric (Data Encryption Standard – DES, Triple DES, Advanced Encryption Standard, Blowfish and Twofish) and asymmetric (RSA) encryption algorithms
  - How data compression, encryption and hashing are used in the storage and transfer of data

- Symbolise, analyse and evaluate
  - Caesar, Polyalphabetic (e.g. Vigenere and Gronsfield), and one-time pad encryption algorithms

- Symbolise and explain
  - Secure data transmission techniques and processes, including the use of encryption, decryption, authentication, hashing and checksums

Resources

Data Encryption Standard (DES)

Data encryption standard (DES) has been found vulnerable against very powerful attacks and therefore, the popularity of DES has been on a slight decline. DES is a block cipher and encrypts data in blocks of size of 64 bits each. This means 64 bits of plain text goes as the input to DES, which produces 64 bits of cipher text. The same algorithm and key are used for encryption and decryption, with minor differences. The key length is 56 bits. DES uses a 56-bit key to encrypt any plain text which can easily be cracked using modern technologies. To prevent this from happening double DES and triple DES were introduced which are much more secure than the original DES because it uses 112-bit and 168-bit keys respectively, offering much more security than DES.


Advanced Encryption Standard

The Advanced Encryption Standard (AES) is found to be at least six times faster than triple DES. AES is an iterative rather than Feistel cipher. AES performs all its computations on
bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix.

**Source:**
https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm

**Blowfish**
Blowfish is a symmetric encryption algorithm, meaning that it uses the same secret key to both encrypt and decrypt messages. Blowfish is also a block cipher, meaning that it divides a message up into fixed length blocks during encryption and decryption. The block length for Blowfish is 64 bits; messages that aren't a multiple of eight bytes in size must be padded.

**Source:** https://www.embedded.com/encrypting-data-with-the-blowfish-algorithm/
**Source:** https://www.geeksforgeeks.org/blowfish-algorithm-with-examples/

**Gronsfeld cipher**
Gronsfeld is essentially a Vigenere cipher, but uses numbers instead of letters. So, a Gronsfeld key of 0123 is the same as a Vigenere key of ABCD.

**Source:** http://rumkin.com/tools/cipher/gronsfeld.php

**One Time Pad Encryption**
In cryptography, a one-time pad is a system in which a private key generated randomly is used only once to encrypt a message that is then decrypted by the receiver using a matching one-time pad and key. Messages encrypted with keys based on randomness have the advantage that there is theoretically no way to "break the code" by analysing a succession of messages. Each encryption is unique and bears no relation to the next encryption so that some pattern can be detected.

**Source:** https://searchsecurity.techtarget.com/definition/one-time-pad

**Twofish**
Twofish is an encryption algorithm. It is a symmetric key block cipher with a block size of 128 bits, with keys up to 256 bits. Twofish has some distinctive features that set it apart from most other cryptographic protocols. It uses pre-computed, key-dependent S-boxes. An S-box (substitution-box) is a basic component of any symmetric key algorithm which performs substitution.

**Source:** https://choosetoencrypt.com/tech/twofish-encryption/

**Vigenère Cipher**
Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets. The encryption of the original text is done using the Vigenère square or Vigenère table. The table consists of the alphabets written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet. At different
points in the encryption process, the cipher uses a different alphabet from one of the rows. The alphabet used at each point depends on a repeating keyword.

arrow Source: https://www.geeksforgeeks.org/vigenere-cipher/

Data Compression, Encryption, Symmetric & Asymmetric Encryption, Hashing

arrow https://pmt.physicsandmathstutor.com/download/Computer-Science/A-level/Notes/OCR/1.3-Exchanging-Data/Advanced/1.3.1.%20Compression,%20Encryption%20and%20Hashing.pdf

Cryptography

arrow See Appendix B

Password Encryption explained

arrow See Appendix D

A Comparison of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish for Guessing Attacks Prevention

arrow https://symbiosisonlinepublishing.com/computer-science-technology/computerscience-information-technology32.php

Protecting data through encryption


Activities

Caesar Cypher worksheet

arrow http://csunplugged.mines.edu/Activities/Cryptography/Cryptography.pdf

Cryptography Activities

arrow https://sites.psu.edu/cvclab/outreach/cryptography-activities-for-k-12/

Hash Function Activity

arrow https://docs.google.com/document/pub?id=1mOPxjuo_ITfhL_cf78nGyg9Eo9SajiPaDR0jiKAnDU8

Brut Force Activity

arrow https://docs.google.com/document/pub?id=1n-8KmbLHxnsB9ZafuIAc9qZ5uX5vP6Uj3Wbs0ysJQA
RSA for encryption activity

➤ https://docs.google.com/document/pub?id=18zYaA-VrUwa82lyfCEw9bHMBqc_eqkVLZF6sX8Scl4g

Cryptography Activities


Information Hiding Activity

➤ https://classic.csunplugged.org/information-hiding/

Practice Exam Questions

The following exam questions are from the mock sample external assessment documents available to the public on the QCAA website.


**QUESTION ONE**

In data security, a hash function can be described as a:

A. One-way cryptographic algorithm that takes an input message of arbitrary length and produces a fixed-length digest

B. Cryptographic algorithm that performs the encryption or decryption of an input message of arbitrary length for output or storage

C. Feature that dynamically negotiates a mutual set of security requirements between two hosts that are attempting to communicate with each other

D. Process of masking password user input with hashes, asterisks or other special characters

---

**QUESTION TWO**

In security systems, it is common practice to use a unique identifier of a numeric type, for example, a door lock code or PIN. People often store these numbers in an unsecured text file on personal devices.

A simple encryption method for a two-digit number is shown below.
Therefore, the Plaintext number ‘45’ would output Ciphertext ‘%\^’.

a) Develop a low-fidelity prototype solution key for any four-digit PIN (e.g. 7826) with associated key and predicted output. The algorithm must read data from a user input and output encrypted data to a file. Annotate the pseudocode in the table below to identify the location of assignment, condition, and modularisation. Evaluate the algorithm using a desk check and write the output at each step.

<table>
<thead>
<tr>
<th>Plaintext number</th>
<th>Block key</th>
<th>Encrypted number</th>
<th>Ciphertext character</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘4’</td>
<td>1 7 3 9 5 8 2 0 4 6 5</td>
<td>%</td>
<td></td>
</tr>
<tr>
<td>‘5’</td>
<td>1 2 3 4 5 6 7 8 9 0 6</td>
<td>^</td>
<td></td>
</tr>
</tbody>
</table>

Therefore, the Plaintext number ‘45’ would output Ciphertext ‘%\^’.

a) Develop a low-fidelity prototype solution key for any four-digit PIN (e.g. 7826) with associated key and predicted output. The algorithm must read data from a user input and output encrypted data to a file. Annotate the pseudocode in the table below to identify the location of assignment, condition, and modularisation. Evaluate the algorithm using a desk check and write the output at each step.

<table>
<thead>
<tr>
<th>Original</th>
<th>Key</th>
<th>Encrypted number</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
b) Compare this encryption method with a standard Caesar cipher, explaining three advantages and/or disadvantages of each method. Explain which of these two methods is the more secure method to use.

QUESTION THREE
Many symmetric encryption algorithms, including the Data Encryption Standard (DES), use a structure commonly referred to as a Feistel network. Feistel networks have a symmetric structure used in the construction of block ciphers. The structure of a Feistel network is best described as a:

A. Cipher that uses public and private keys for secure encryption across networks
B. Cipher that uses a process of character substitution to encrypt or decrypt the same block of data
C. Very similar or identical set of iterative structural processes to encrypt or decrypt the same block of data
D. Structural process where plain text digits are combined with a stream of pseudorandom cipher digits
Concept Two Focus: Useability Principles & Elements and Principles of Visual Communication

**Objectives**

**Topic One**

- **Recognise and describe**
  - How useability principles are used to inform solution development
  - How the elements and principles of visual communication inform user interface development

**Resources**

- **Useability Principles**

- **Elements and Principles of Visual Design**

**Activities**

**Activity 1:**

List 5 sites or apps with good UI design, and 1-2 with bad design. For each, articulate in your own words why you believe it works or does not work.

**Activity 2:**

Copy the UI of a good site/app pixel-for-pixel. At the end, write down specific techniques the creator used that “expand your design vocabulary”.

**Activity 3:**

Record any fonts you particularly like, noting characteristics like when they seem to work well and where you have seen them.
Activity 4:
Create 2-3 style tiles for fonts in your database that you particularly want to experiment with. Think of a concept for a site, then design out a style tile featuring:

- Realistic font choices and text elements (e.g. headers, navigational elements, body text)
- Appropriate colour palette
- Example form controls or imagery
- Logo (optional)

Activity 5:
Drawing from the style tiles you have created in Activity 4, design 1-3 screens of an app/website that you are most interested in working on.

Practice Exam Questions

The following exam questions are from the mock sample external assessment documents available to the public on the QCAA website.


QUESTION ONE

A website’s content and functionality was evaluated using the four criteria:

- **Operable**, e.g. the website should be navigable without a mouse and with a screen reader
- **Perceivable**, e.g. images should have text equivalents
- **Robust**, e.g. the website should work with different devices and navigation should be consistent
- **Understandable**, e.g. text should not be more complicated than it needs to be and the website should operate in a predictable manner

Identify the useability principle tested by the four criteria above.

A. Effectiveness
B. Accessibility
C. Learnability
D. Utility
QUESTION TWO
Describe two elements of visual communication that can be implemented to enhance the user experience.
Concept Three Focus: Security of Data

Objectives

Topic One
- Explain
  - Australian Privacy Principles (2014) and ethics applicable to the use of personally identifiable or sensitive data from a digital systems perspective

Topic Two
- Analyse problems and information to determine
  - Factors and risks that affect data security, including confidentiality, integrity and availability, and privacy
- Analyse and evaluate
  - Data to ensure completeness, consistency and integrity

Resources

Australian Privacy

The Open University: Introduction to Cyber Security Free course
- https://www.futurelearn.com/courses/introduction-to-cyber-security

What is Cybersecurity?

Sample assessment resources from the QCAA website.

Activities

Data Security Test
- https://www.bbc.co.uk/bitesize(guides/zw3cwmn/revision/3
Concept Four Focus: Data Flow Diagrams

Objectives

Topic One
- Symbolise
  - Representations of a digital solution
  - Data flow through a system using data flow diagrams
- Describe
  - Data using appropriate naming conventions, data formats and structures

Topic Two
- Analyse and explain
  - a system’s data process by developing data flow diagrams that link external entities, data sources, processes and data storage

Resources

Data Flow Diagrams
QCAA supporting document.
- Appendix C

Data Flow Diagram – Accessing / Using an Email Server
- http://www.youtube.com/watch?v=ZFlynt3K3U0

Context Diagram (Sample 1)
- http://www.youtube.com/watch?v=hiMeEswjWuk&t

Context & Data Flow Diagrams (Sample 2)
- http://www.youtube.com/watch?v=X-O6s5sah4o

Context & Data Flow Diagrams (Sample 3)
- http://www.youtube.com/watch?v=dFb21Bldf0A

DFD Symbols and Diagrams
The same as the book.
- http://www.youtube.com/watch?v=i3lgsdefgkU
Why do we need DFD’s?
⇒ http://www.youtube.com/watch?v=kBeUY8noj6A

Rewrite DFD examples to suit our DFD symbols
⇒ http://www.youtube.com/watch?v=6VGTVgaJiLM

Rules for creating DFD’s
⇒ On page 125 of textbook

Student DFD Example
⇒ https://www.youtube.com/watch?v=Ik85hZkyYPA

Activities

Possible Data Flow Diagram Activity
⇒ http://www.youtube.com/watch?v=VeLkVD0Q_6M

Practice Exam Questions

The following exam questions are from the mock sample external assessment documents available to the public on the QCAA website.


Questions begin over page
QUESTION ONE
Examine the following data flow diagram for a business that makes and sells widgets.

From the diagram, which of the following statements is most correct?
A. Production receives input based on other processes and entities.
B. This business consists of four main processes and three entities.
C. The sale process receives input from multiple sources.
D. Employees provide information to support the sale.

QUESTION TWO
A medical practice is developing a system for their patient bookings and record keeping. A centralised database store will be implemented with access available from several medical practices across the city.

The system will contain a variety of non-encrypted patient information (e.g. names and contact details), patient bookings, encrypted patient medical records and doctor information. Doctors will be able to enter and edit all data, including medical records. Employees of each practice will be able to enter and edit patient information and bookings, but not medical records. Patients will be able to enter and update their own patient information and enter and view their own bookings.

a) Symbolise the practice’s networked database requirements in a high-level data flow diagram. The data flow diagram has been started for you below.
b) Explain two key security features required in this system and justify why you would include them.
Concept Five Focus: Algorithms

**Objectives**

**Topic One**
- **Recognise and describe**
  - How simple algorithms consist of input, process and output at various stages
- **Symbolise and explain**
  - The basic constructs of an algorithm, including assignment, sequence, selection, condition, iteration and modularisation

**Topic Two**
- **Symbolise**
  - Algorithmic steps using pseudocode

**Topic Three**
- **Evaluate**
  - By desk checking algorithms to predict the output for a given input, identify errors and validate algorithms

**Resources**

**Algorithmic Design Method**
QCAA supplement document.
- see Appendix A

**What is an algorithm?**
- [https://youtu.be/6hfOvs8pY1k](https://youtu.be/6hfOvs8pY1k)

**Fundamentals of algorithms**
- [https://www.bbc.co.uk/bitesizeguides/zjddqhv/revision/1](https://www.bbc.co.uk/bitesizeguides/zjddqhv/revision/1)

**Standard algorithms (search and sort algorithms)**
- [https://www.bbc.co.uk/bitesizeguides/z7kww6f/revision/6](https://www.bbc.co.uk/bitesizeguides/z7kww6f/revision/6)

**Data Types and Structures**
- [https://www.bbc.co.uk/bitesizeguides/z788jty/revision/1](https://www.bbc.co.uk/bitesizeguides/z788jty/revision/1)
INPUT and OUTPUT validation

→ https://www.bbc.co.uk/bitesize/guides/zfnny4j/revision/1

File handling (open, close, read, write)

→ https://www.bbc.co.uk/bitesize/guides/zfnny4j/revision/10

Fibonacci


Digital Solutions Textbook

→ Page 38 – 43

Activities

Activity 1: Test on Algorithms / Pseudocode

→ https://www.bbc.co.uk/bitesize/guides/z3bq7ty/test

Activity 2: Algorithm Activities

→ https://classic.csunplugged.org/searching-algorithms/

Activity 3: Challenge

Write an algorithm for the following problem (see page 42 of your textbook for a hint count algorithm).

Design logic in pseudocode for a program that asks the user for a number and outputs all the numbers added together (and including) the number they provided.

e.g. User enters 3 - program returns 6
(0+ 1 + 2 + 3 = 6)

Step 1: Write an algorithm

Step 2: Copy the algorithms to VSC and translate to Javascript program

Solution over page
**Activity 3: Solution**

**START**

NUM userInput  
NUM counter = 0  
NUM storage = 0

INPUT userInput  
WHILE counter <= userInput  
    storage = storage + counter  
    ADD 1 to count  
ENDWHILE

OUTPUT storage

**END**

**Activity 4: Challenge**

In a simple number game the program generates a secret number between 1 and 100. In no more than 10 guesses you try to guess the number. After each guess the program tells you if your guess was too high, too low or correct. The program also keeps track of how many guesses you have had and tells you the game is over when you use all ten of your guesses or when you guess the number correctly.

Write an algorithm (pseudocode) to describe the processes run by the program to play the game. In your solution include:

- A condition which checks for illegal guesses (those less than 1 or greater than 100).
- Function (method) to generate a secret number
- Function to check the guess and provide appropriate feedback

This solution will require use of the following control structures **sequence, iteration, selection** and **functions**.

Which of the following iteration methods would you choose? Why?:

A. Counted loop (FOR loop)  
B. Pre-tested loop (WHILE loop)  
C. Post-tested loop (REPEAT___ UNTIL loop)

Review some **control structures** on the following sites to help you with the solution:
WRITE THE ALGORITHM - adhere to the pseudocode rules.

Activity 4: Solution
An algorithm to describe a game in which the user tries to guess a number between 1 and 100, using no more than ten guesses.

BEGIN Program
    SET NumOfGuess to 0 (or number of guesses is set to 0)
    SET GotIt to FALSE

    GENERATE a secret number using random number function/method
    REPEAT
        GET a guess from the user
        IF the guess is in range THEN
            INCREMENT the number of guesses
            Check the guess
        ELSE
            WRITE the guess is out of range
        ENDIF
    UNTIL guess is correct (GotIt is TRUE) OR number of guesses is 10
    IF the guess is incorrect (GotIt is FALSE) THEN
        WRITE you have run out of guesses (=10)
        SHOW the secret number
    ENDIF
END Program

BEGIN FUNCTION check the guess
    IF guess > secret number THEN
        WRITE the guess is too big (or tell the user their guess is too big)
    ELSE
        IF guess < secret number THEN
            WRITE the guess is too small
        ELSE
        ENDIF
END FUNCTION
ELSE

WRITE Congratulations, you got it
SHOW the number of guesses it took to get the secret number
SET GotIt to TRUE
ENDIF
ENDIF
END FUNCTION check the guess

Practice Exam Questions

The following exam questions are from the mock sample external assessment documents available to the public on the QCAA website.


QUESTION ONE

The following algorithm is used to produce an output based on different input data values.

BEGIN

IF value1 > value2

PRINT "A"
ELSE

IF value3 ≤ (value1 + value2)

PRINT "B"
ELSE

PRINT "C"
ENDIF
ENDIF
IF (value3 – value1) ≥ value2

PRINT "D"
ENDIF
END

Question continued over page
Identify the output produced by the algorithm when value 1 = 7, value 2 = 8 and value 3 = 15.

A. AD  
B. BC  
C. CD  
D. BD

QUESTION TWO

The table below shows two algorithms.

<table>
<thead>
<tr>
<th>Program 1</th>
<th>Program 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGIN Main</td>
<td>BEGIN Main</td>
</tr>
<tr>
<td>num1 = 50</td>
<td>num1 is global</td>
</tr>
<tr>
<td>Sub()</td>
<td>num1 = 50</td>
</tr>
<tr>
<td>PRINT num1</td>
<td>Sub()</td>
</tr>
<tr>
<td>END Main</td>
<td>PRINT num1</td>
</tr>
<tr>
<td>BEGIN Sub</td>
<td>END Main</td>
</tr>
<tr>
<td>num1 is local</td>
<td>BEGIN Sub</td>
</tr>
<tr>
<td>num1 = 20</td>
<td>num1 = 20</td>
</tr>
<tr>
<td>END Sub</td>
<td>END Sub</td>
</tr>
</tbody>
</table>

Select the correct output when both programs are run.

<table>
<thead>
<tr>
<th>Program 1</th>
<th>Program 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. 50 20</td>
<td>20 20</td>
</tr>
<tr>
<td>B. 50 50</td>
<td>50 50</td>
</tr>
<tr>
<td>C. 20 50</td>
<td>50 20</td>
</tr>
<tr>
<td>D. 20 20</td>
<td>20 20</td>
</tr>
</tbody>
</table>
QUESTION THREE

The two algorithms below are for the 'fizzbuzz' test. The algorithms evaluate numbers between 1 and 100 and print:

- ‘fizz’ if the number is a factor of 3
  OR
- ‘buzz’ if the number is a factor of 5
  OR
- ‘fizbuzz’ if the number is a factor of both 3 and 5
  OR
- The original number is the number is factor of neither 3 nor 5.

<table>
<thead>
<tr>
<th>Algorithm 1</th>
<th>Algorithm 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGIN</td>
<td>BEGIN</td>
</tr>
<tr>
<td><em>FOR</em> i = 1 to 100</td>
<td><em>i = 1</em></td>
</tr>
<tr>
<td>IF i modulus 3 = 0</td>
<td>WHILE i &lt;= 100</td>
</tr>
<tr>
<td><em>PRINT</em> “fizz”</td>
<td>IF i modulus 3 = 0</td>
</tr>
<tr>
<td><em>ENDIF</em></td>
<td>PRINT_ “fizz”</td>
</tr>
<tr>
<td>IF i modulus 5 = 0</td>
<td>IF i modulus 5 = 0</td>
</tr>
<tr>
<td><em>PRINT</em> “buzz”</td>
<td>PRINT_ “buzz”</td>
</tr>
<tr>
<td><em>ENDIF</em></td>
<td>IF NOT i modulus 3 = 0</td>
</tr>
<tr>
<td>IF i modulus 5 = 0 and i modulus 3 = 0</td>
<td>AND NOT i modulus 5 = 0</td>
</tr>
<tr>
<td><em>PRINT</em> “fizbuzz”</td>
<td>PRINT_ i</td>
</tr>
<tr>
<td>ELSE</td>
<td>PRINT_ new line</td>
</tr>
<tr>
<td><em>PRINT</em> i</td>
<td>i = i + 1</td>
</tr>
<tr>
<td><em>ENDIF</em></td>
<td><em>ENDWHILE</em></td>
</tr>
<tr>
<td><em>PRINT</em> new line</td>
<td></td>
</tr>
<tr>
<td><em>NEXT</em></td>
<td></td>
</tr>
<tr>
<td><em>END</em></td>
<td><em>END</em></td>
</tr>
</tbody>
</table>

The algorithms were evaluated against the criteria of accuracy and reliability.

Examine the algorithms and choose the statement that is true.

A. Algorithm 1 is more reliable than Algorithm 2.
B. Algorithm 2 is more reliable than Algorithm 1.
C. Algorithm 1 is more accurate than Algorithm 2.
D. Algorithm 2 is more accurate than Algorithm 1.
Concept Six Focus: Pseudocode

Objectives

Topic Two
- Symbolise algorithmic steps as pseudocode

Resources

How to write Pseudocode
- https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/

Common Action Keywords
- http://users.csc.calpoly.edu/~jdalbey/SWE/pdl_std.html

Activities

Activity 1:
Each statement in your pseudocode should express just one action for the computer. If the task list is properly drawn, then in most cases each task will correspond to one line of pseudocode.

Exercise:
- Read name, hourly rate, hours worked, deduction rate
- Perform calculations gross = hourlyRate * hoursWorked
deduction = grossPay * deductionRate
- net pay = grossPay – deduction
- Write name, gross, deduction, net pay

Solution over page
Activity 1: Solution Pseudocode

BEGIN

READ name, hourlyRate, hoursWorked, deductionRate

grossPay = hourlyRate * hoursWorked

deduction = grossPay * deductionRate

netPay = grossPay – deduction

WRITE name, grossPay, deduction, netPay

END

Activity 2:

Write a short pseudocode program that will print out the multiplication table up to 10 for a given input value. For example, if I used the input value of 3 my output would be:

1 x 3 = 3
2 x 3 = 6
3 x 3 = 9
4 x 3 = 12
5 x 3 = 15
6 x 3 = 18
7 x 3 = 21
8 x 3 = 24
9 x 3 = 27
10 x 3 = 30

Practice Exam Questions

The following exam questions are from the mock sample external assessment documents available to the public on the QCAA website.


Questions begin over page
QUESTION 1

The following algorithm produces an output from input data.

Main program
BEGIN
  var1 = “4”
  var3 = 3
  process3()
  process2(var3)
  process1()
END

Sub_process1 ()
BEGIN
  PRINT var1 + “4”
Sub END

Sub_process2 (var2)
BEGIN
  PRINT 4 + var2
Sub END

Sub_process3 ()
BEGIN
  PRINT var3 / 3
Sub END

The output of the pseudocode will be:

A. 1744
B. 1544
C. 178
D. 158
QUESTION TWO
Refer to Stimulus 1 below.

Stimulus 1
BEGIN
  INPUT Answer INTEGER
  INPUT Guess INTEGER

  SET Correct = FALSE
  SET Loop = 0
  REPEAT
    INPUT Answer = “Player 1 enter a number between 1 and 10.”
    UNTIL Answer > 1 OR Answer < 10

  DO WHILE NOT Correct AND Loop < 4
    INPUT Guess = “Player 2, enter your guess.”
    IF Guess = Answer THEN
      SET Correct = TRUE
      OUTPUT “Well done Player 2!”
    ELSE
      IF Guess < Answer THEN
        OUTPUT “Too high”
      ELSE
        OUTPUT “Too low”
      END IF
    ELSE
      END IF
    END IF
    SET Loop = Loop + 1
  END WHILE

  OUTPUT “The number was:” Answer
END

Question continued over page
Pseudocode has been developed for a two-player ‘guess the number’ game. The game is played when:

- Player 1 inputs a number ranging from and including 1 to 10
- Player 2 attempts to guess the number entered by Player 1. Player 2 has three attempts to guess the correct number.

a) The pseudocode contains errors. Analyse the pseudocode to identify and explain each error. Using the same variables, make refinements to the pseudocode to correct the identified errors.

b) Desk check your corrected algorithm by completing the test table below. Include any WRITE statements that are generated during testing.

Assume Player 1 inputs the number 5, Player 2 is not able to correctly guess the answer during any of their attempts, and all Player 2 inputs are valid.

<table>
<thead>
<tr>
<th>Player 1 input</th>
<th>Loop</th>
<th>Guess</th>
<th>Value of ‘Correct’ at Line 21</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
QUESTION THREE

Use pseudocode to write an algorithm to determine whether a word contains two consecutive letters that are the same.

Examples of words that contain two consecutive letters that are the same as ‘ball’, ‘moon’ and ‘possible’.

__________________________________________________________________________________________

__________________________________________________________________________________________

__________________________________________________________________________________________

__________________________________________________________________________________________

__________________________________________________________________________________________
Concept Seven Focus: SQL Querying

Objectives

Topic One
- Analyse problems and information to determine
  - Requirements of the solution requirements

Resources

Learn SQL
- https://youtu.be/nWyyDHhTxYU

SQL Join Types Explained
- Appendix E

SQL Interactive Lessons
- https://sqlbolt.com/lesson/introduction

Activities

Activity 1: SQL Teaching
- https://www.sqlteaching.com/#!select

Activity 2: SQL Tutorial
- https://sqlzoo.net/

Activity 3: SQL Online Quiz
- https://www.w3schools.com/sql/sql_quiz.asp
**QUESTION ONE**

The following extract from a medical database shows patient, vaccines by item number, when data vaccination is due, date immunised and the risk of contracting the disease post-immunisation.

<table>
<thead>
<tr>
<th>Patients</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>First</td>
</tr>
<tr>
<td>1684</td>
<td>John</td>
</tr>
<tr>
<td>2893</td>
<td>Mary</td>
</tr>
<tr>
<td>3642</td>
<td>Jackie</td>
</tr>
<tr>
<td>1206</td>
<td>Michael</td>
</tr>
<tr>
<td>1910</td>
<td>Midori</td>
</tr>
<tr>
<td>1934</td>
<td>Beatrice</td>
</tr>
<tr>
<td>1935</td>
<td>Wanda</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Assess</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Vaccine</td>
<td>Risk</td>
</tr>
<tr>
<td>1</td>
<td>Measles</td>
<td>0.001</td>
</tr>
<tr>
<td>2</td>
<td>Polio</td>
<td>0.00021</td>
</tr>
<tr>
<td>3</td>
<td>Tetanus</td>
<td>0.0005</td>
</tr>
</tbody>
</table>

```sql
SELECT first, last, immunised
FROM results r
INNER JOIN assess a ON r.item = a.item
INNER JOIN patients p ON p.id = r.id
WHERE a.item in (1,2)
AND r.immunised < a.due ;
```

The query above determines

A. The first name, last name and immunisation date for all patients who received items before they were due.

B. The first name, last name and immunisation date for patients who received Items 1 and 2 after they were due.
C. The first name, last name and immunisation date for patients who received Item 1 or Item 2 before it was due.

D. The first name, last name and immunisation date for patients who received Items 1 and 2 before they were due.

**QUESTION TWO**

Refer to Stimulus 2 below.

**Stimulus 2**

<table>
<thead>
<tr>
<th>Students</th>
<th>ID</th>
<th>First</th>
<th>Last</th>
</tr>
</thead>
<tbody>
<tr>
<td>871</td>
<td>Hans Schmidt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>862</td>
<td>Bill White</td>
<td></td>
<td></td>
</tr>
<tr>
<td>869</td>
<td>Nguyen Ng</td>
<td></td>
<td></td>
</tr>
<tr>
<td>854</td>
<td>Ann Devlin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>831</td>
<td>Hans Manabeim</td>
<td></td>
<td></td>
</tr>
<tr>
<td>872</td>
<td>Betty Thompson</td>
<td></td>
<td></td>
</tr>
<tr>
<td>868</td>
<td>Will Fletcher</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Assess</th>
<th>Item</th>
<th>Description</th>
<th>Weight</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Presentation</td>
<td>10</td>
<td>0809</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Assignment</td>
<td>30</td>
<td>2110</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Exam</td>
<td>60</td>
<td>0211</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Results</th>
<th>Item</th>
<th>ID</th>
<th>Submitted</th>
<th>Mark (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>871</td>
<td>0809</td>
<td>80</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>862</td>
<td>0709</td>
<td>60</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>854</td>
<td>0809</td>
<td>70</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>872</td>
<td>1009</td>
<td>55</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>868</td>
<td>0609</td>
<td>90</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>869</td>
<td>0909</td>
<td>70</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>871</td>
<td>2110</td>
<td>70</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>869</td>
<td>2110</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>872</td>
<td>2110</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>862</td>
<td>2110</td>
<td>70</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>868</td>
<td>2110</td>
<td>75</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>869</td>
<td>0211</td>
<td>95</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>872</td>
<td>1012</td>
<td>45</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>862</td>
<td>0812</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>868</td>
<td>3011</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>871</td>
<td>0712</td>
<td>60</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>854</td>
<td>1112</td>
<td>65</td>
</tr>
</tbody>
</table>

Stimulus 2 is an extract from a university course database that shows enrolled students, assessment items and student results for each assessment item.

The following query was developed to show all students (by ID) and their overall subject result.

```
SELECT ID, SUM(Mark / 100)
FROM Results, Assess
GROUP BY Item
```
a) Evaluate the query and identify and describe any errors.

______________________________________________________________________________________________________________________________

______________________________________________________________________________________________________________________________

______________________________________________________________________________________________________________________________

______________________________________________________________________________________________________________________________

b) Refine the query to ensure it functions correctly. Justify the refinements made.

______________________________________________________________________________________________________________________________

______________________________________________________________________________________________________________________________

______________________________________________________________________________________________________________________________

______________________________________________________________________________________________________________________________

c) Refine the query to show only students with a subject rank of 'A'.

______________________________________________________________________________________________________________________________

______________________________________________________________________________________________________________________________

______________________________________________________________________________________________________________________________

______________________________________________________________________________________________________________________________
Concept Eight Focus: Desk Checking

Objectives

Topic Three
- Evaluate
  - by desk checking algorithms to predict the output for a given input, identify errors and validate algorithms

Resources

What is desk checking?
Desk checking is a way of testing your algorithm; it can be also called as code walk through.

Source: https://www.professionalqa.com/desk-checking

Desk checking algorithms

https://youtu.be/i2qLAVBUERs

Activities

Activity 1:
Desk Checking Algorithm Activities

Activity 2:
The following pseudocode algorithm calculates the area of a floor so that a carpet can be fitted that is the correct size.

Perform a desk-check on this algorithm using a table to show the values in each variable after the execution of each line. You should assume an input value of 50 for length and 50 for width.

1. data width as whole number
2. data length as whole number
3. data area as whole number
4. output “Welcome to the carpet area calculator”
5. output “What is the width of your floor”
6. input width
7. output “What is the length of your floor”
8. input length
9. area = length * width
10. output “For your floor you will need a carpet that is:"
11. output area
12. output “Thankyou for using this program.”
Concept Nine Focus: Networking terminology explained

Objectives

Topic One

- **Symbolise and explain**
  - How application sub-systems e.g. front end, back end, work together to constitute a solution

- **Explain**
  - Network transmission principles, including latency, jitter, guarantee and timeliness of delivery, and protocols relevant to the transmission of data over the internet e.g. HTTP, HTTPS, FTP, VPN, streaming and broadcasting data packets
  - Methods for data exchange used to transfer data across networked systems including REST, JSON and XML

Resources

**The Internet Crash Course**
Covering network types, IP & how data travels over the network.

→ [https://youtu.be/AEaKrq3SpW8](https://youtu.be/AEaKrq3SpW8)

**World Wide Web**

→ [https://youtu.be/guvsH5OFizE](https://youtu.be/guvsH5OFizE)

**Khan Academy "How the Internet Works"**


**Khan Academy “The Internet”**
Covers wide range of concepts about file transfer protocols and security, readings, and quizzes.


**FTP explained**

→ [https://www.youtube.com/watch?v=tOj8MSEIbfA](https://www.youtube.com/watch?v=tOj8MSEIbfA)
What is a VPN?

⇒ https://www.youtube.com/watch?v=q4P4BijXghQ

REST API essentials

⇒ https://restfulapi.net/security-essentials/

Remote APIs

⇒ https://www.youtube.com/watch?v=GZvSYJDk-us&t=775s

How the web works

⇒ https://www.youtube.com/watch?v=GZvSYJDk-us&t=1024s

RESTful API Constraint Scavenger Hunt

⇒ https://www.youtube.com/watch?v=GZvSYJDk-us&t=1320s

Web API security

⇒ https://www.imperva.com/learn/application-security/web-api-security/

REST API essentials

⇒ https://restfulapi.net/security-essentials/

How SMTP mail server works

⇒ https://www.hostinger.com/tutorials/how-to-use-free-google-smtp-server

JSON and XML differences and syntax


Stepped through introduction to JSON

⇒ https://restfulapi.net/introduction-to-json/

Activities

Activity 1: HTTP and Abstraction on the Internet

⇒ https://curriculum.code.org/csp-19/unit1/13/

Activity 2: Network Security Activities

⇒ https://groklearning.com/course/cyber-910-py-networking/

Activity 3: Coding with JSON and JavaScript basics

Practice Exam Questions

The following exam questions are from the mock sample external assessment documents available to the public on the QCAA website.


QUESTION ONE

JSON uses human-readable text to transmit data objects.

{  “name”: “John”}
{  “age”: 30}
{  “employees”: [“John”, “Anna”, “Peter”]}
{  “sale”: true}

The JSON data types in the code above would be declared as:

A. John, 30, John, Sale
B. name, age, employees, sale
C. char, number, list, true/false
D. string, integer, array of strings, Boolean

QUESTION TWO

An IP address can be exchanged with a(n):

A. MIME-type
B. Point-to-Point Protocol
C. Domain name
D. Email address
E. Usenet group

Questions continued over page
QUESTION THREE
Which of the following is a valid IP address:
A. 192.168.1.1
B. www.apple.com
C. activa@midcoast.com
D. http://www.pages.net/index.html
E. 12 Dreamcatcher Way, Hope, ME 04847

QUESTION FOUR
When a background image is smaller than the viewport, by default the browser will:
A. simply ignore the background
B. Display the background in the upper-left corner with the background colour filling the rest of the window
C. Tile the background image to fill the window
D. Stretch the background image to fill the window
E. Crash

QUESTION FIVE
HTTP stands for:
A. High Technology Transmission Protocol
B. Help Text Translation Protocol
C. Hypertext Transfer Protocol
D. Hardware Testing Tool Protocol
E. How to Talk Protocol

Questions continued over page
QUESTION SIX

Briefly describe how a browser requests a URL, how the URL is handled, and how the server responds, in terms of the DNS system, encoding schemes and data protocols.

Question Six Response:

The IP address is broken down into packets by the Transmission Control Protocol. After a packet is created by the TCP, the Internet Protocol (IP) then takes over and sends the packet to its destination along a route. There are domain name servers who converts text-based domain name addresses into real IP addresses. Each time a user connects to a URL, the domain name server is consulted, and the destination address is converted to numbers.
APPENDICES

APPENDIX A: Algorithms

Digital Solutions 2019 v1.2
Supporting resource: Representing algorithms with pseudocode

Purpose
The purpose of this resource is to provide supporting information to the syllabus requirements for Digital Solutions 2019.

Syllabus subject matter

Algorithmic design method
Pseudocode will be used as the formal method of representing algorithms in this syllabus. Pseudocode is a descriptive method used to represent an algorithm and is a mixture of everyday language and programming conventions.

Pseudocode implements the basic control structures of assignment, sequence, selection, condition, iteration and modularisation through the use of keywords associated with the constructs, and textual indentation. Used to show how a computing algorithm should and could work, it is often an intermediate step in programming between the planning stage and writing executable code.

Pseudocode can also be useful for:
- demonstrating thinking that later can become comments in the final program
- describing how an algorithm should work
- explaining a computing process to less technical people
- generating code in collaboration with others.

Pseudocode does not have a standard format and varies from programmer to programmer. However, a number of conventions are generally used.
Conventions for writing pseudocode

KEYWORDS are written in bold capitals and are often words taken directly from programming languages. For example, IF, THEN and ELSE are all words that can be validly used in most languages. OUTPUT and COMPUTE are from the language COBOL and WRITE is from the language Pascal. Keywords do not have to be valid programming language words as long as they clearly convey the intent of the line of pseudocode.

Statements that form part of a REPETITION LOOP are indented by the same amount to indicate that they form a logical grouping.

In a similar way, IF, THEN and ELSE statements are indented to clearly distinguish the alternative processing paths.

The end of REPETITION LOOPS and IF, THEN and ELSE statements are explicitly indicated by the use of ENDDO and ENDF at the appropriate points.

Pseudocode should clearly indicate what is happening at each step, including formulas of calculations. For example:

```
CALCULATE net is not as clear as CALCULATE net = gross - tax.
```

Programmers prefer to use a more abbreviated version in which memory cells used to store the input are given program-like names. For example:

```
INPUT num1
INPUT num2
```

is preferable to

```
INPUT first number
INPUT second number
```

See: Subject matter in the Digital Solutions 2019 syllabus


Further considerations

Digital Solutions 2019 subject matter describes conventions for writing pseudocode (above). While these are not exhaustive, additional information outlined in the tables that follow is used when providing students with learning opportunities.
### Additional considerations for writing pseudocode

#### Language
Common keywords are written in bold capitals. Keywords do not have to be valid programming language words as long as they clearly convey the intent of the line of pseudocode.

Statements in a block are indented by the same amount to show hierarchy.

#### Naming convention
Use camel case naming convention for variables, subroutines, methods and functions.

#### Modularity
Pseudocode always starts and ends with the **BEGIN** and **END** keywords.

- **Main algorithm:** Procedures, subroutines, methods or functions:
  - BEGIN name
  - statements
  - END name

#### Variables
Programmers use names without spaces for variables. In pseudocode, this will make the algorithm.

- **INPUT** num1 is preferable to **INPUT** FirstNumber
- **INPUT** num2 is preferable to **INPUT** SecondNumber

To input, assign or output values, common words can be used as keywords.

For example:

- **INPUT** mark
- **WRITE** "the total is" count
- **PRINT** x, y
- **DISPLAY** name, result
- **READ** name from list.txt
- **OUTPUT** average

#### Assignment
Pseudocode should clearly indicate what is happening at each step. For example:

- **CALCULATE** net = gross - tax is clearer than **CALCULATE** net

#### Selection
A control structure used for decisions or branching and choosing alternate paths.

The beginning and end of these structures are indicated with keywords (for multiple branches).

- **IF** condition **THEN**
  - statements
- **ENDIF**

- **ELSE**
  - statements
- **ENDIF**

#### Iterations (loops)
Control structures to provide repetitions. There are three main types of loops.

Each has a clear start and end, with the statements within the loop indented.

- **FOR** post-test loops:
  - **REPEAT**
  - statements
  - **UNTIL** condition

- **FOR** pre-test loops:
  - **WHILE** condition
  - statements

- **FOR** counted loops:
  - **FOR** count = startVal **TO** endVal
  - statements

Other statement types and other constructs can be represented in similar ways.

#### Font
A mono-space typeface, such as Courier New, is recommended when writing algorithms on computer.

Vertical quotation marks should also be used. e.g. " and .
## Supplementary explanations

The following explanations also provide support for teaching and learning.

<table>
<thead>
<tr>
<th>Term</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>efficiency</td>
<td>a situation in which a system or machine uses minimal resources such as time and processing power while still achieving its goals. There are two types algorithmic and code efficiency. Algorithmic efficiency refers to the reliability, speed and programming methodology for developing succinct structures within an application. Code efficiency is directly linked with algorithmic efficiency and the speed of runtime execution for software. It is the key element in ensuring high performance. The goal of code efficiency is to reduce resource consumption and completion time as much as possible with minimum risk to the business or operating environment. The software product quality can be evaluated using algorithm or code used efficiency.</td>
</tr>
<tr>
<td>maintainability</td>
<td>easy to read code, that is easy to dissect so parts relating to a required change is easy to modify without risking a chain reaction of errors in dependant modules</td>
</tr>
<tr>
<td>reliability of software or hardware</td>
<td>attribute of software (data and algorithms) or hardware that consistently performs without failure and according to its specifications</td>
</tr>
<tr>
<td>technical specification</td>
<td>a set of requirements that a product must meet or exceed. In Digital Solutions: Program specifications describe what the software is required to achieve. Functional specifications describe the manner in which the program specifications are achieved. These specifications may be regarded as prescribed criteria.</td>
</tr>
</tbody>
</table>
14 CAESAR CIPHER

Source: Free book - Invent your own games with Python

This chapter is a resource not to be replicated with code.

The program in this chapter isn’t really a game, but it is fun nevertheless. The program will convert normal English into a secret code. It can also convert secret codes back into regular English. Only someone who knows the key to the secret codes will be able to understand the messages.

Because this program manipulates text to convert it into secret messages, you’ll learn several new functions and methods for manipulating strings. You’ll also learn how programs can do math with text strings just as they can with numbers.

TOPICS COVERED IN THIS CHAPTER
• Cryptography and ciphers
• Ciphertext, plaintext, keys, and symbols
• Encrypting and decrypting
• The Caesar cipher
• The find() string method
• Cryptanalysis
• The brute-force technique

Cryptography and Encryption
The science of writing secret codes is called cryptography. For thousands of years, cryptography has made it possible to send secret messages that only the sender and
recipient could read, even if someone captured the messenger and read the coded message. A secret code system is called a cipher. The cipher used by the program in this chapter is called the Caesar cipher.

In cryptography, we call the message that we want to keep secret the plaintext. Let’s say we have a plaintext message that looks like this:

There is a clue behind the bookshelf.

Converting the plaintext into the encoded message is called encrypting the plaintext. The plaintext is encrypted into the ciphertext. The ciphertext looks like random letters, so we can’t understand what the original plaintext was just by looking at the ciphertext. Here is the previous example encrypted into ciphertext:

aolyl pz h jsBl ilopuk Aol ivvrzolsm.

If you know the cipher used to encrypt the message, you can decrypt the ciphertext back to the plaintext. (Decryption is the opposite of encryption.)

Many ciphers use keys, which are secret values that let you decrypt ciphertext that was encrypted with a specific cipher. Think of the cipher as being like a door lock. You can only unlock it with a particular key.

If you’re interested in writing cryptography programs, you can read my book Hacking Secret Ciphers with Python. It’s free to download from http://inventwithpython.com/hacking/.

How the Caesar Cipher Works

The Caesar cipher was one of the earliest ciphers ever invented. In this cipher, you encrypt a message by replacing each letter in it with a “shifted” letter. In cryptography, the encrypted letters are called symbols because they can be letters, numbers, or any other signs. If you shift the letter A by one space, you get the letter B. If you shift the letter A by two spaces, you get the letter C. Figure 14-1 shows some letters shifted by three spaces.

![Figure 14-1: A Caesar cipher shifting letters by three spaces. Here, B becomes E.](image-url)
To get each shifted letter, draw a row of boxes with each letter of the alphabet. Then draw a second row of boxes under it but start your letters a certain number of spaces over. When you get to the end of the plaintext alphabet, wrap back around to A. Figure 14-2 shows an example with the letters shifted by three spaces.

![Figure 14-2: The entire alphabet shifted by three spaces](image)

The number of spaces you shift your letters (between 1 and 26) is the key in the Caesar cipher. Unless you know the key (the number used to encrypt the message), you won’t be able to decrypt the secret code. The example in Figure 14-2 shows the letter translations for the key 3.

**Note:**
While there are 26 possible keys, encrypting your message with 26 will result in a ciphertext that is exactly the same as the plaintext!

If you encrypt the plaintext word HOWDY with a key of 3, then:

- The letter H becomes K.
- The letter O becomes R.
- The letter W becomes Z.
- The letter D becomes G.
- The letter Y becomes B.

So, the ciphertext of HOWDY with the key 3 becomes KRZGB. To decrypt KRZGB with the key 3, we go from the bottom boxes back to the top.

If you would like to include lowercase letters as distinct from uppercase letters, then add another 26 boxes to the ones you already have and fill them with the 26 lowercase letters. Now with a key of 3, the letter Y becomes b, as shown in Figure 14-3.
The cipher works the same way as it did with just uppercase letters. In fact, if you want to use letters from another language’s alphabet, you can write boxes with those letters to create your cipher.

**Sample Run of Caesar Cipher**

Here is a sample run of the Caesar Cipher program encrypting a message:

Do you wish to encrypt or decrypt a message?

**encrypt**

Enter your message:

The sky above the port was the color of television, tuned to a dead channel.

Enter the key number (1-52)

13

Your translated text is:

gur FxL noBlr Gur CBEG JnF Gur pByBE Bs GryrlvFvBA, GHArq GB n qrnq punAAry.

Now run the program and decrypt the text that you just encrypted:

Do you wish to encrypt or decrypt a message?

**decrypt**

Enter your message:

gur FxL noBlr Gur CBEG JnF Gur pByBE Bs GryrlvFvBA, GHArq GB n qrnq punAAry.

Enter the key number (1-52)

13

Your translated text is:

The sky above the port was the color of television, tuned to a dead channel.

If you do not decrypt with the correct key, the text will not decrypt properly:

Do you wish to encrypt or decrypt a message?

**decrypt**
Enter your message:

Enter the key number (1-52)

15

Your translated text is:

Rfc qiw YZmtc rfc nmpr uYq rfc amjmp md rcjctggml, rslcb rm Y bcYb afYIcj.

Source Code for Caesar Cipher

Enter this source code for the Caesar Cipher program and then save the file as cipher.py.

If you get errors after entering this code, compare the code you typed to the book’s code with the online diff tool at https://www.nostarch.com/inventwithpython#diff.

cipher.py
1. # Caesar Cipher
2. SYMBOLS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'
3. MAX_KEY_SIZE = len(SYMBOLS)
4. def getMode():
5.     while True:
6.         print('Do you wish to encrypt or decrypt a message?')
7.         mode = input().lower()
8.         if mode in ['encrypt', 'e', 'decrypt', 'd']:
9.             return mode
10.        else:
12. print('Enter either "encrypt" or "e" or "decrypt" or "d".
13.
14. def getMessage():
15.     print('Enter your message:')
16.     return input()
17.
18. def getKey():
19.     key = 0
20.     while True:
21.         print('Enter the key number (1-%s) % (MAX_KEY_SIZE))
22.         key = int(input())
23.         if (key >= 1 and key <= MAX_KEY_SIZE):
24.             return key
25.
26. def getTranslatedMessage(mode, message, key):
27.     if mode[0] == 'd':
28.         key = -key
29.     translated = "
30.     for symbol in message:
31.         symbolIndex = SYMBOLS.find(symbol)
32.         if symbolIndex == -1: # Symbol not found in SYMBOLS.
33.             # Just add this symbol without any change.
34.             translated += symbol
35.         else:
36.             # Encrypt or decrypt.
37.             symbolIndex += key
38.             if symbolIndex >= len(SYMBOLS):
39.                 symbolIndex -= len(SYMBOLS)
40.             elif symbolIndex < 0:
41.                 symbolIndex += len(SYMBOLS)
42.             translated += SYMBOLS[symbolIndex]
43.     return translated
44.
45.
48. mode = getMode()
49. message = getMessage()
50. key = getKey()
51. print('Your translated text is: ')
52. print(getTranslatedMessage(mode, message, key))

Setting the Maximum Key Length
The encryption and decryption processes are the reverse of each other, but they share much of the same code. Let's look at how each line works:

1. # Caesar Cipher
2. SYMBOLS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'
3. MAX_KEY_SIZE = len(SYMBOLS)
MAX_KEY_SIZE is a constant that stores the length of SYMBOLS (52). This constant reminds us that in this program, the key used in the cipher should always be between 1 and 52.

Deciding to Encrypt or Decrypt the Message
The getMode() function lets the user decide whether they want to use the program's encryption or decryption mode:

5. def getMode():
6.     while True:
7.         print('Do you wish to encrypt or decrypt a message?')
8.         mode = input().lower()
9.         if mode in ['encrypt', 'e', 'decrypt', 'd']:
10.            return mode
11.        else:
12.            print('Enter either "encrypt" or "e" or "decrypt" or "d"."

Line 8 calls input() to let the user enter the mode they want. The lower() method is then called on this string to return a lowercase version of the string. The value returned from input().lower() is stored in mode. The if statement's condition checks whether the string stored in mode exists in the ['encrypt', 'e', 'decrypt', 'd'] list.

This function will return the string in mode as long as mode is equal to 'encrypt', 'e', 'decrypt', or 'd'. Therefore, getMode() will return the string mode. If the user types something that is not 'encrypt', 'e', 'decrypt', or 'd', then the while loop will ask them again.
Getting the Message from the Player

The `getMessage()` function simply gets the message to encrypt or decrypt from the user and returns it:

```python
14. def getMessage():
15.     print('Enter your message:')
16.     return input()
```

The call for `input()` is combined with `return` so that we use only one line instead of two.

Getting the Key from the Player

The `getKey()` function lets the player enter the key they will use to encrypt or decrypt the message:

```python
18. def getKey():
19.     key = 0
20.     while True:
21.         print('Enter the key number (1-%s)' % (MAX_KEY_SIZE))
22.         key = int(input())
23.         if (key >= 1 and key <= MAX_KEY_SIZE):
24.             return key
```

The while loop ensures that the function keeps looping until the user enters a valid key. A valid key here is one between the integer values 1 and 52 (remember that `MAX_KEY_SIZE` is 52 because there are 52 characters in the `SYMBOLS` variable). The `getKey()` function then returns this key. Line 22 sets `key` to the integer version of what the user entered, so `getKey()` returns an integer.

Encrypting or Decrypting the Message

The `getTranslatedMessage()` function does the actual encrypting and decrypting:

```python
26. def getTranslatedMessage(mode, message, key):
27.     if mode[0] == 'd':
28.         key = -key
29.     translated = ''
```

It has three parameters:

- `mode` This sets the function to encryption mode or decryption mode.
- `message` This is the plaintext (or ciphertext) to be encrypted (or decrypted).
- `key` This is the key that is used in this cipher.

Line 27 checks whether the first letter in the `mode` variable is the string 'd'. If so, then the program is in decryption mode. The only difference between decryption and encryption mode is that in decryption mode, the key is set to the negative version of itself. For example, if `key` is the
integer 22, then decryption mode sets it to -22. The reason is explained in “Encrypting or Decrypting Each Letter” on page 205.

The translated variable will contain the string of the result: either the ciphertext (if you are encrypting) or the plaintext (if you are decrypting). It starts as a blank string and has encrypted or decrypted characters concatenated to the end of it. Before we can start concatenating the characters to translated, however, we need to encrypt or decrypt the text, which we'll do in the rest of getTranslatedMessage().

Finding Passed Strings with the find() String Method

In order to shift the letters around to do the encryption or decryption, we first need to convert them into numbers. The number for each letter in the SYMBOLS string will be the index where it appears. Since the letter A is at SYMBOLS[0], the number 0 will represent the capital A. If we wanted to encrypt this with the key 3, we would simply use 0 + 3 to get the index of the encrypted letter: SYMBOLS[3] or 'D'.

We'll use the find() string method, which finds the first occurrence of a passed string in the string on which the method is called. Enter the following in the interactive shell:

```python
>>> 'Hello world!'.find('H')
0
>>> 'Hello world!'.find('o')
4
>>> 'Hello world!'.find('ell')
1
```

'Hello world!'.find('H') returns 0 because the 'H' is found at the first index of the string 'Hello world!'. Remember, indexes start at 0, not 1. The code 'Hello world!'.find('o') returns 4 because the lowercase 'o'is first found at the end of 'Hello'. The find() method stops looking after the first occurrence, so the second 'o' in 'world' doesn't matter. You can also find strings with more than one character. The string 'ell' is found starting at index 1.

If the passed string cannot be found, the find() method returns -1:

```python
>>> 'Hello world!'.find('xyz')
-1
```

Let's go back to the Caesar Cipher program. Line 31 is a for loop that iterates on each character in the message string:

```python
31.     for symbol in message:
32.         symbolIndex = SYMBOLS.find(symbol)
33.         if symbolIndex == -1: # Symbol not found in SYMBOLS.
34.             # Just add this symbol without any change.
35.             translated += symbol
```

The find() method is used on line 32 to get the index of the string in symbol. If find() returns -1, the character in symbol will just be added to translated without any change. This means that any characters that aren't part of the alphabet, such as commas and periods, won't be changed.
Encrypting or Decrypting Each Letter

Once you’ve found a letter’s index number, adding the key to the number will perform the shift and give you the index for the encrypted letter.

Line 38 does this addition to get the encrypted (or decrypted) letter.

37.             # Encrypt or decrypt.
38.             symbolIndex += key

Remember that on line 28, we made the integer in key negative for decryption. The code that adds the key will now subtract it, since adding a negative number is the same as subtraction.

However, if this addition (or subtraction, if key is negative) causes symbolIndex to go past the last index of SYMBOLS, we’ll need to wrap around to the beginning of the list at 0. This is handled by the if statement starting at line 40:

40.             if symbolIndex >= len(SYMBOLS):
41.                 symbolIndex -= len(SYMBOLS)
42.             elif symbolIndex < 0:
43.                 symbolIndex += len(SYMBOLS)
44.             translated += SYMBOLS[symbolIndex]

Line 40 checks whether symbolIndex has gone past the last index by comparing it to the length of the SYMBOLS string. If it has, line 41 subtracts the length of SYMBOLS from symbolIndex. If symbolIndex is now negative, then the index needs to wrap around to the other side of the SYMBOLS string. Line 42 checks whether the value of symbolIndex is negative after adding the decryption key. If so, line 43 adds the length of SYMBOLS to symbolIndex.

The symbolIndex variable now contains the index of the correctly encrypted or decrypted symbol. SYMBOLS[symbolIndex] will point to the character for this index, and this character is added to the end of translated on line 45.

The execution loops back to line 31 to repeat this for the next character in message. Once the loop is done, the function returns the encrypted (or decrypted) string in translated on line 46:

46.     return translated

The last line in the getTranslatedMessage() function returns the translated string.

Starting the Program

The start of the program calls each of the three functions defined previously to get the mode, message, and key from the user:

48. mode = getMode()
49. message = getMessage()
50. key = getKey()
51. print('Your translated text is:')[
52. print(getTranslatedMessage(mode, message, key))

These three values are passed to getTranslatedMessage(), whose return value (the translated string) is printed to the user.

Expanding the symbols

If you want to encrypt numbers, spaces, and punctuation marks, just add them to the SYMBOLS string on line 2. For example, you could have your cipher program encrypt numbers, spaces, and punctuation marks by changing line 2 to the following:

2. SYMBOLS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz 1234567890!@#$%^&*()'

Note that the SYMBOLS string has a space character after the lowercase z.

If you wanted, you could add even more characters to this list. And you don’t need to change the rest of your program, since all the lines of code that need the character list just use the SYMBOLS constant.

Just make sure that each character appears only once in the string. Also, you’ll need to decrypt your message with the same SYMBOLS string that it was encrypted with.

The Brute-Force Technique

That’s the entire Caesar cipher. However, while this cipher may fool some people who don’t understand cryptography, it won’t keep a message secret from someone who knows cryptanalysis. While cryptography is the science of making codes, cryptanalysis is the science of breaking codes.

The whole point of cryptography is to make sure that if someone else gets their hands on the encrypted message, they cannot figure out the original text. Let’s pretend we are the code breaker and all we have is this encrypted text:

LwCjBA uiG vwB jm xtmiAivB, jCB kmzBiqvBG qA ijACzl.

Brute-forcing is the technique of trying every possible key until you find the correct one. Because there are only 52 possible keys, it would be easy for a cryptanalyst to write a hacking program that decrypts with every possible key. Then they could look for the key that decrypts to plain English. Let’s add a brute-force feature to the program.

Adding the Brute-Force Mode

First, change lines 7, 9, and 12—which are in the getMode() function—to look like the following (the changes are in bold):

5. def getMode():
6.     while True:
7.         print('Do you wish to encrypt or decrypt or brute-force a
message?

8. mode = input().lower()
9. if mode in ['encrypt', 'e', 'decrypt', 'd', 'brute', 'b']:
10. return mode
11. else:
12. print("Enter either "encrypt" or "e" or "decrypt" or "d" or "brute" or "b".")

This code will let the user select brute force as a mode.

Next, make the following changes to the main part of the program:

48. mode = getMode()
49. message = getMessage()
50. if mode[0] != 'b':
51. key = getKey()
52. print('Your translated text is:')
53. if mode[0] != 'b':
54. print(getTranslatedMessage(mode, message, key))
55. else:
56. for key in range(1, MAX_KEY_SIZE + 1):
57. print(key, getTranslatedMessage('decrypt', message, key))

If the user is not in brute-force mode, they are asked for a key, the original getTranslatedMessage() call is made, and the translated string is printed.

However, if the user is in brute-force mode, then the getTranslatedMessage() loop iterates from 1 all the way up to MAX_KEY_SIZE (which is 52). Remember that the range() function returns a list of integers up to, but not including, the second parameter, which is why we add + 1. The program will then print every possible translation of the message (including the key number used in the translation). Here is a sample run of this modified program:

Do you wish to encrypt or decrypt or brute-force a message?

brute

Enter your message:
LwCjBA uiG vwB jm xtmIAivB, jCB kmzBiqvBG qA iJACzl.
Your translated text is:
1 KvBiAz thF uvA il wshzhuA, iBA jlyAhpuAF pz hizByk.
2 JuAhzy sgE tuz hk vrkgygtz, hAz ikxzgotzE oy ghyAxj.
3 Itzgyx rFD sty gj uqjfxfsy, gzy hjwvfnSyD nx fgxzwi.
4 Hsyfxw qeC rsx fi tpiewerx, fyx givxemrxC mw efwyvh.
5 Grxewv pdB qrw eh sohdvdqw, exw fhuwlqiwB lv devxug.
6 Fqwvdvu oCA pqv dg mngucpv, dww egtvckpvA ku cduwtf.
7 Epvcut nbz opu cf qmfbtbou, cvu dfsubjouz jt bctvse.
8 Doubts may not be pleasant, but certainty is absurd.
9 Cntasr lIZx mns ad okdZrZms, ats bdqsZhmsx hr Zartqc.
10 BmsZrq kYw lmYr Zc njcYqYlr, Zsr acprYglrw gq YZqsqb.
11 AlrYqp jXv klq Yb mibXpXkq, Yrq ZboqXfkq fp XYproa.
12 zkqXpo iWu jkp Xa IhaWoWjp, Xqp YanpWejpu eo WXoqnZ.

--snip--

After looking over each row, you can see that the eighth message isn't nonsense but plain English! The cryptanalyst can deduce that the original key for this encrypted text must have been 8. This brute-force method would have been difficult to do back in the days of Julius Caesar and the Roman Empire, but today we have computers that can quickly go through millions or even billions of keys in a short time.

Summary

Computers are good at doing math. When we create a system to translate some piece of information into numbers (as we do with text and ordinals or with space and coordinate systems), computer programs can process these numbers quickly and efficiently. A large part of writing a program is figuring out how to represent the information you want to manipulate as values that Python can understand.

While our Caesar Cipher program can encrypt messages that will keep them secret from people who have to figure them out with pencil and paper, the program won’t keep them secret from people who know how to get computers to process information. (Our brute-force mode proves this.)

In Chapter 15, we’ll create Reversegam (also known as Reversi or Othello). The AI that plays this game is much more advanced than the AI that played Tic-Tac-Toe. In fact, it’s so good that most of the time you won’t be able to beat it!
APPENDIX C: Data Flow Diagrams

Digital Solutions 2019 v1.2
Supporting resource: Using data flow diagrams

Purpose
The purpose of this resource is to provide supporting information to the syllabus requirements for Digital Solutions 2019.

Syllabus subject matter

Process-oriented analysis methods
Data flow diagrams (DFD), which include data source, data flow, data storage and process, are used to represent system interrelationships, data, system or process-oriented workflow.

DFDs are graphical representations of data flow through an information system. They do not represent programming logic or processing steps. Data flow symbols are used to represent data source, flow, storage and processes (Figure 4).

Figure 4: Data flow diagram symbols, their names and functions

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name and function</th>
<th>Symbol</th>
<th>Name and function</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Data source or External entity symbol]</td>
<td>Data source or External entity — a source or destination of data flow that is outside the area of study</td>
<td>![Data flow symbol]</td>
<td>Data flow — a connector shows relationships between the representative shapes</td>
</tr>
<tr>
<td>![Data store symbol]</td>
<td>Data store — repository of data; “D” indicates a permanent computer file; “M” indicates a manual file; “T” indicates a transient store, deleted after processing</td>
<td>![Level symbol]</td>
<td>Process — transforms incoming data flow into outgoing data flow</td>
</tr>
</tbody>
</table>

See: Subject matter in the Digital Solutions 2019 syllabus

Further considerations

Digital Solutions 2019 subject matter describes conventions for writing pseudocode (above). While these are not exhaustive, additional information outlined in the tables that follows may also be useful when providing students with learning opportunities.
**Figure 1: Data flow diagram symbols, their names, functions and conventions**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name, functions and conventions</th>
<th>Symbol</th>
<th>Name, functions and conventions</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Data source or External entity" /></td>
<td>Data source or External entity — a source or destination of data flow that is outside the area of study. These elements: • must connect to at least one process • are usually labelled with nouns.</td>
<td><img src="image" alt="Data flow" /></td>
<td>Data flow — a connector shows relationships between the representative shapes. These connectors: • must travel in only one direction and to or from a process • must not cross each other • are usually labelled with nouns and describe information not actions.</td>
</tr>
<tr>
<td><img src="image" alt="Datastore" /></td>
<td>Datastore — repository of data. These elements are labelled with: • a ‘D’ to indicate a permanent computer file • an ‘M’ to indicate a manual file • a ‘T’ to indicate a transient store which is deleted after processing. These elements are usually labelled with nouns.</td>
<td><img src="image" alt="Process" /></td>
<td>Process — transforms incoming data flow into outgoing data flow. These elements: • must have at least one input flow and one output flow • must transform data • are numbered for identification purposes • are usually labelled with verbs.</td>
</tr>
</tbody>
</table>
## Supplementary explanations

The following explanations also provide support for teaching and learning.

<table>
<thead>
<tr>
<th>Term</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>external data</td>
<td>data that is from a source external to the digital solution. It includes flat files, databases managed by a DBMS, data streams or logs collected from remote sensors and media collections.</td>
</tr>
<tr>
<td>internal data</td>
<td>data that is part of the digital solution. It includes data loaded in memory, entered by the user through the user interface or collected by local sensors. It may also include local files or databases created by coded components of the digital solution.</td>
</tr>
<tr>
<td>data structure</td>
<td>a particular way of organizing data in a computer so that it can be used effectively. The aim is to reduce the space and time complexities of different tasks.</td>
</tr>
</tbody>
</table>
APPENDIX D: Encrypted Password Explained

Encrypting password

Using md5(); function to make your login system more secure.

Syntax

```php
$password="123456";
md5($password);
```

Use md5(); to encrypts password to make it more secure

Overview

Look at these two databases, it's the same person and same info, the first one we don't encrypt his password, but the second one we encrypted his password.

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>email</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John Smith</td>
<td><a href="mailto:john@somewhere.com">john@somewhere.com</a></td>
<td>john856</td>
</tr>
</tbody>
</table>

When you encrypt "john856" using this code, you'll see the result "ad65d5054042fda44ba3f3c97ce80c6"

This is not a random result, every time you encrypt the same password you will get the same result.

```php
$password="john856";
$encrypt_password=md5($password);
echo $encrypt_password;
```

Example - Login

This is an example Login with encrypted password, but don't forget to encrypt password and insert into database when your user sign up.

```php
// username and password sent from form
$myusername=$_POST["myusername"]; $mypassword=$_POST["mypassword"];

// encrypt password
$encrypted_mypassword=md5($mypassword);

$sql="SELECT * FROM $tbl_name WHERE username='$myusername' and password='$encrypted_mypassword'";
$result=mysql_query($sql);
```

You can learn to create login system here.
APPENDIX E: SQL Join Types Explained

UNIT 4 DIGITAL SOLUTIONS SYLLABUS INSERT:

SQL CREATE, DROP and ALTER statements - SQL INSERT and UPDATE - SQL SELECT query, including WHERE, GROUP BY, HAVING, ORDER BY, sub-selection and inner-joins clauses

SQL INNER JOIN Keyword

Source: https://www.sqltutorial.org/sql-inner-join/

The most important and frequently used of the joins is the INNER JOIN. They are also referred to as an EQUIJOIN.
Source: https://www.tutorialspoint.com/sql/sql-inner-joins.htm

The INNER JOIN keyword selects records that have matching values in both tables.
Source: https://www.w3schools.com/sql/sql_join_inner.asp

What is Inner Join in SQL?

The INNER JOIN selects all rows from both participating tables as long as there is a match between the columns. An SQL INNER JOIN is same as JOIN clause, combining rows from two or more tables. The INNER JOIN in SQL joins two tables according to the matching of a certain criteria using a comparison operator.

INNER JOIN Syntax

SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;

Below is a selection from the "Orders" table:

<table>
<thead>
<tr>
<th>OrderID</th>
<th>CustomerID</th>
<th>EmployeeID</th>
<th>OrderDate</th>
<th>ShipperID</th>
</tr>
</thead>
<tbody>
<tr>
<td>10308</td>
<td>2</td>
<td>7</td>
<td>1996-09-18</td>
<td>3</td>
</tr>
<tr>
<td>10309</td>
<td>37</td>
<td>3</td>
<td>1996-09-19</td>
<td>1</td>
</tr>
<tr>
<td>10310</td>
<td>77</td>
<td>8</td>
<td>1996-09-20</td>
<td>2</td>
</tr>
</tbody>
</table>

And a selection from the "Customers" table:

<table>
<thead>
<tr>
<th>CustomerID</th>
<th>CustomerName</th>
<th>ContactName</th>
<th>Address</th>
<th>City</th>
<th>PostalCode</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alfreds Futterkiste</td>
<td>Maria Anders</td>
<td>Obere Str. 57</td>
<td>Berlin</td>
<td>12209</td>
<td>Germany</td>
</tr>
</tbody>
</table>
SQL INNER JOIN Example

The following SQL statement selects all orders with customer information:

```sql
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
INNER JOIN Customers
ON Orders.CustomerID = Customers.CustomerID;
```

**Example:** SQL INNER JOIN between two tables

Here is an example of inner join in SQL between two tables.

**Sample table: foods**

```
+---------+--------------+-----------+------------+
| ITEM_ID | ITEM_NAME    | ITEM_UNIT | COMPANY_ID |
+---------+--------------+-----------+------------+
| 1       | Chex Mix     | Pcs       | 16         |
| 6       | Cheez-It     | Pcs       | 15         |
| 2       | BN Biscuit   | Pcs       | 15         |
| 3       | Mighty Munch | Pcs       | 17         |
| 4       | Pot Rice     | Pcs       | 15         |
| 5       | Jaffa Cakes  | Pcs       | 18         |
| 7       | Salt n Shake | Pcs       |            |
+---------+--------------+-----------+------------+
```

**Sample table: company**

```
+------------+---------------+--------------+
| COMPANY_ID | COMPANY_NAME  | COMPANY_CITY |
+------------+---------------+--------------+
| 18         | Order All     | Boston       |
| 15         | Jack Hill Ltd | London       |
+------------+---------------+--------------+
```
To join item name, item unit columns from foods table and company name, company city columns from company table, with the following condition:

- company_id of foods and company table must be same

the following SQL statement can be used:

**SQL Code:**

```sql
SELECT foods.item_name, foods.item_unit,
    company.company_name, company.company_city
FROM foods
INNER JOIN company
ON foods.company_id = company.company_id;
```

**Output:**

<table>
<thead>
<tr>
<th>ITEM_NAME</th>
<th>ITEM_UNIT</th>
<th>COMPANY_NAME</th>
<th>COMPANY_CITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chex Mix</td>
<td>Pcs</td>
<td>Akas Foods</td>
<td>Delhi</td>
</tr>
<tr>
<td>Cheez-It</td>
<td>Pcs</td>
<td>Jack Hill Ltd</td>
<td>London</td>
</tr>
<tr>
<td>BN Biscuit</td>
<td>Pcs</td>
<td>Jack Hill Ltd</td>
<td>London</td>
</tr>
<tr>
<td>Mighty Munch</td>
<td>Pcs</td>
<td>Foodies.</td>
<td></td>
</tr>
<tr>
<td>Pot Rice</td>
<td>Pcs</td>
<td>Jack Hill Ltd</td>
<td>London</td>
</tr>
<tr>
<td>Jaffa Cakes</td>
<td>Pcs</td>
<td>Order All</td>
<td>Boston</td>
</tr>
</tbody>
</table>

**Example of SQL INNER JOIN using JOIN keyword**

To get item name, item unit columns from foods table and company name, company city columns from company table, after joining these mentioned tables, with the following condition:

- company id of foods and company id of company table must be same,

the following SQL statement can be used:

**SQL Code:**

```sql
SELECT foods.item_name, foods.item_unit,
    company.company_name, company.company_city
FROM foods
JOIN company
ON foods.company_id = company.company_id;
```
FROM foods
JOIN company
ON foods.company_id = company.company_id;

Output:

<table>
<thead>
<tr>
<th>ITEM_NAME</th>
<th>ITEM_ COMPANY_NAME</th>
<th>COMPANY_CITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chex Mix</td>
<td>Pcs</td>
<td>Akas Foods</td>
</tr>
<tr>
<td>Cheez-It</td>
<td>Pcs</td>
<td>Jack Hill Ltd</td>
</tr>
<tr>
<td>BN Biscuit</td>
<td>Pcs</td>
<td>Jack Hill Ltd</td>
</tr>
<tr>
<td>Mighty Munch</td>
<td>Pcs</td>
<td>Foodies.</td>
</tr>
<tr>
<td>Pot Rice</td>
<td>Pcs</td>
<td>Jack Hill Ltd</td>
</tr>
</tbody>
</table>

SQL INNER JOIN for all columns

To get all the columns from foods and company table after joining, with the following condition:

- company id of foods and company id of company table must be same

The following SQL statement can be used:

SQL Code:

SELECT *
FROM foods
JOIN company
ON foods.company_id = company.company_id;

Output:

<table>
<thead>
<tr>
<th>ITEM_ID</th>
<th>ITEM_NAME</th>
<th>ITEM_</th>
<th>COMPANY_NAME</th>
<th>COMPANY_CITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chex Mix</td>
<td>Pcs</td>
<td>Akas Foods</td>
<td>Delhi</td>
</tr>
<tr>
<td>6</td>
<td>Cheez-It</td>
<td>Pcs</td>
<td>Jack Hill Ltd</td>
<td>London</td>
</tr>
<tr>
<td>2</td>
<td>BN Biscuit</td>
<td>Pcs</td>
<td>Jack Hill Ltd</td>
<td>London</td>
</tr>
<tr>
<td>3</td>
<td>Mighty Munch</td>
<td>Pcs</td>
<td>Foodies.</td>
<td>London</td>
</tr>
<tr>
<td>4</td>
<td>Pot Rice</td>
<td>Pcs</td>
<td>Jack Hill Ltd</td>
<td>London</td>
</tr>
<tr>
<td>5</td>
<td>Jaffa Cakes</td>
<td>Pcs</td>
<td>Order All</td>
<td>London</td>
</tr>
</tbody>
</table>
**Difference between JOIN and INNER JOIN**

JOIN returns all rows from tables where the key record of one table is equal to the key records of another table.

The **INNER JOIN** selects all rows from both participating tables as long as there is a match between the columns. An SQL INNER JOIN is same as JOIN clause, combining rows from two or more tables.

An inner join of A and B gives the result of A intersect B, i.e. the inner part of a Venn diagram intersection.

Inner joins use a comparison operator to match rows from two tables based on the values in common columns from each table. For example, retrieving all rows where the student identification number is the same for both the students and courses tables.

**Using JOIN Clause**

```
SELECT * FROM Table1 JOIN Table2
ON Table1.column_name=Table2.column_name;
```

**Using INNER JOIN Clause**

```
SELECT *
FROM Table1 INNER JOIN Table2
ON Table1.column_name= Table2.column_name;
```

**Difference between INNER JOIN and OUTER JOIN**

An **INNER JOIN** is such type of join that returns all rows from both the participating tables where the key record of one table is equal to the key records of another table. This type of join required a comparison operator to match rows from the participating tables based on a common field or column of both the tables.

The **OUTER JOIN** returns all rows from the participating tables which satisfy the condition and also those rows which do not match the condition will appear in this operation. This result set can appear in three types of format -

The first one is LEFT OUTER JOIN, in this join includes all the rows from a left table of JOIN clause and the unmatched rows from a right table with NULL values for selected columns.

The second one is RIGHT OUTER JOIN, in this join includes all rows from the right of JOIN cause and the unmatched rows from the left table with NULL values for selected columns.

The last one in FULL OUTER JOIN, in this join, includes the matching rows from the left and right tables of JOIN clause and the unmatched rows from left and right table with NULL values for selected columns.
Example:
Here is two table \textit{tableX} and \textit{tableY} and they have no duplicate rows in each. In \textit{tableX} the values (A,B) are unique and in \textit{tableY} the values (E,F) are unique, but the values (C and D) are common in both the tables.

```
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>C</td>
<td>E</td>
</tr>
<tr>
<td>D</td>
<td>F</td>
</tr>
</tbody>
</table>
```

Here is \texttt{INNER JOIN}

```
SELECT *
FROM tableX
INNER JOIN tableY on tableX.X = tableY.Y;
```

or

```
SELECT tableX.*,tableY.*
FROM tableX,tableY
WHERE tableX.X = tableY.Y;
```

Output:

```
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
</tr>
</tbody>
</table>
```

Here only the matching of both \textit{tableX} and \textit{tableY} have appeared in the result set.

Here is \texttt{LEFT OUTER JOIN}

```
SELECT tableX.*,tableY.*
FROM tableX,tableY
WHERE tableX.X = tableY.Y(+)
```
or

```
SELECT *
FROM tableX
LEFT OUTER JOIN tableY ON tableX.X= tableY.Y
```

**Output:**

Here all the rows from `tableX` that is left side of JOIN clause and all the rows with NULL values for unmatched columns from `tableY` that is the right side of JOIN clause have appeared.

**Here is RIGHT OUTER JOIN**

```
SELECT * FROM tableX
RIGHT OUTER JOIN tableY ON tableX.X= tableY.Y
```

**Output:**

Here all the rows from `tableY` that is the right side of JOIN clause and all the rows with NULL values for unmatched columns from `tableX` that is left side of JOIN clause have appeared.
Here is FULL OUTER JOIN

```
SELECT *
FROM tableX
FULL OUTER JOIN tableY ON tableX.X = tableY.Y
```

Output:

```
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>NULL</td>
</tr>
<tr>
<td>B</td>
<td>NULL</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>NULL</td>
<td>E</td>
</tr>
<tr>
<td>NULL</td>
<td>F</td>
</tr>
</tbody>
</table>
```

Here all the matching rows from `tableX` and `tableY` and all the unmatched rows with NULL values for both the tables have appeared.

**INNER JOIN ON vs WHERE clause**

The WHERE clause, what is done is that all records that match the WHERE condition are included in the result set but an INNER JOIN is that, data not matching the JOIN condition is excluded from the result set.

Linking between two or more tables should be done using an INNER JOIN ON clause but filtering on individual data elements should be done with WHERE clause.

INNER JOIN is ANSI syntax whereas the WHERE syntax is more relational model oriented.

The INNER JOIN is generally considered more readable and it is a cartesian product of the tables, especially when you join lots of tables but the result of two tables JOIN'ed can be filtered on matching columns using the WHERE clause.
# Glossary

**NB:** These terms are not included in the Unit 4 in the Digital Solutions syllabus glossary.

<table>
<thead>
<tr>
<th>TERM</th>
<th>DEFINITION</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>ASCII stands for the &quot;American Standard Code for Information Interchange&quot;. ASCII is a 7-bit character set containing 128 characters. It contains the numbers from 0-9, the uppercase and lowercase English letters from A to Z, and some special characters. The character-sets used in modern computers, HTML, and Internet are all based on ASCII.</td>
<td><a href="https://w3schools.sinsixx.com/tags/ref_ascii.asp.htm">https://w3schools.sinsixx.com/tags/ref_ascii.asp.htm</a></td>
</tr>
<tr>
<td>Authentication</td>
<td>Verifying the integrity of a transmitted message and/or verifying the identity of a user logging into a network or computer. Passwords, digital certificates, smart cards and biometrics can be used to prove user identity. Digital certificates can be used to identify the network to the client.</td>
<td><a href="https://www.pcmag.com/encyclopedia/term/authentication">https://www.pcmag.com/encyclopedia/term/authentication</a></td>
</tr>
<tr>
<td>Bandwidth</td>
<td>Bandwidth measures how much data can flow through a specific connection at one time. Bandwidth is typically expressed in <em>bits per second</em>, like 60 Mbps or 60 Mb/s, to explain a data transfer rate of 60 million bits (megabits) every second.</td>
<td><a href="https://techterms.com/definition/bandwidth">https://techterms.com/definition/bandwidth</a></td>
</tr>
<tr>
<td>Bitrate</td>
<td>Bitrate describes the rate at which bits are transferred from one location to another. In other words, it measures how much data is transmitted in a given amount of time. Bitrate is commonly measured in bits per second (bps), kilobits per second (Kbps), or megabits per second (Mbps).</td>
<td><a href="https://techterms.com/definition/bitrate">https://techterms.com/definition/bitrate</a></td>
</tr>
</tbody>
</table>
### Checksum
A checksum is a value (sequence of numbers or letters) used to verify the integrity of a file or a data transfer. In other words, it is a sum that checks the validity of data. Checksums are typically used to compare two sets of data to make sure they are the same.

**Source:** [https://techterms.com/definition/checksum](https://techterms.com/definition/checksum)

### Client-Server architecture
The "client-server" architecture is common in both local and wide area networks. For example, if an office has a **server** that stores the company's database on it, the other computers in the office that can access the database are "**clients**" of the server.

**Source:** [https://techterms.com/definition/client](https://techterms.com/definition/client)

### Cybersecurity
Cybersecurity is the set of practices and tools that individuals, IT staff, and governments use to keep information and devices safe from attackers. Ultimately, the goal of cybersecurity is to ensure the integrity, confidentiality, and availability of digital information. Files must be accessible to authorised users on demand, but must remain inaccessible to anyone else.


### Data Confidentiality
In the context of computer systems, allows authorised users to access sensitive and protected data. Specific mechanisms ensure confidentiality and safeguard data from harmful intruders.

**Source:** [https://www.techopedia.com/definition/10254/confidentiality](https://www.techopedia.com/definition/10254/confidentiality)

### Data Input
Any information or data sent to a computer for processing is considered input. Input or user input is sent to a computer using an input device.

**Source:** [https://www.computerhope.com/jargon/i/input.htm](https://www.computerhope.com/jargon/i/input.htm)

### Data Output
Any information that is processed by and sent out from a computer or other electronic device is considered output. An example of output is anything viewed on your computer monitor screen.

**Source:** [https://www.computerhope.com/jargon/o/output.htm](https://www.computerhope.com/jargon/o/output.htm)

### Data Privacy
Data privacy is about authorized access — who has it and who defines it (legal issue).

| **Data Protection** | Data protection is about securing data against unauthorized access (technical issue).  
|---------------------|----------------------------------------------------------------------------------------------------------------------|
| **Data Security**   | Data security refers to protective digital privacy measures that are applied to prevent unauthorized access to computers, databases and websites. Data security also protects data from corruption.  
| **Data Structure**  | The physical layout of data.  
| **Data transmission** | Data transmission is the process of sending digital or analog data over a communication medium to one or more computing, network, communication or electronic devices. It enables the transfer and communication of devices in a point-to-point, point-to-multipoint and multipoint-to-multipoint environment.  
*Source:* [https://www.techopedia.com/definition/9756/data-transmission](https://www.techopedia.com/definition/9756/data-transmission) |
| **Digital Environment** | A context, or a “place”, that is enabled by technology and digital devices, often transmitted over the Internet, or other digital means, e.g., mobile phone network. Records and evidence of an individual's interaction with a digital environment constitute their digital footprint.  
| **Domain Name System (DNS)** | Domain Name System (DNS) is a protocol within the set of standards for how computers exchange data on the internet and on many private networks, known as the TCP/IP protocol suite. It helps convert domain names like "howstuffworks.com" into an Internet Protocol (IP) address, such as 70.42.251.42 that computers use to identify each other on the network.  
*Source:* [https://computer.howstuffworks.com/dns.htm](https://computer.howstuffworks.com/dns.htm) |
| **DNS spoofing** | DNS cache poisoning, also known as DNS spoofing, is a type of attack that exploits vulnerabilities in the domain name system (DNS) to divert Internet traffic away from legitimate servers and towards fake ones.  
### Fibre Optic Cable

Fibre optic cable used to transmit a serial bit stream using pulses of light.

**Source:** [https://erg.abdn.ac.uk/users/gorry/course/phy-pages/fibre.html](https://erg.abdn.ac.uk/users/gorry/course/phy-pages/fibre.html)

### Firewall

A firewall is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules.


### Hashing

Hashing is the practice of taking a string or input key, a variable created for storing narrative data, and representing it with a hash value, which is typically determined by an algorithm and constitutes a much shorter string than the original.

**Source:** [https://www.techopedia.com/definition/14316/hashing-cybersecurity](https://www.techopedia.com/definition/14316/hashing-cybersecurity)

### Internet Protocol (IP)

The Internet Protocol (IP) is part of the TCP/IP protocol suite. IP uses a packet-switched architecture, in which data are broken up into smaller "packets," with each packet containing a source address and destination address. IP packets are handed over to a data link layer protocol, such as Ethernet, for the actual, physical transmission to the next node in the network path.

**Source:** [https://www.pcmag.com/encyclopedia/term/internet-protocol](https://www.pcmag.com/encyclopedia/term/internet-protocol)

### Internet Service Provider (ISP)

An Internet service provider (ISP) is a company that provides customers with Internet access. Data may be transmitted using several technologies, including dial-up, DSL, cable modem, wireless or dedicated high-speed interconnects. Also known as an Internet access provider (IAP)

**Source:** [https://www.techopedia.com/definition/2510/internet-service-provider-isp](https://www.techopedia.com/definition/2510/internet-service-provider-isp)
**Jitter**

Jitter is also referred technically as packet delay variation. This pertains to the variance in time delay in milliseconds (ms) between data packets over a network.

Source: [https://www.speedcheck.org/wiki/jitter/](https://www.speedcheck.org/wiki/jitter/)

---

**Network types**

A **wireless** network allows devices to stay connected to the network but roam untethered to any wires. Access points amplify Wi-Fi signals, so a device can be far from a router but still be connected to the network. A **wired** network uses cables to connect devices, such as laptop or desktop computers, to the Internet or another network.


A **local area network (LAN)** is a collection of devices connected together in one physical location, such as a building, office, or home.

A **wide area network (WAN)** or metropolitan area network (MAN) covers larger geographic areas.


A **cellular mobile network** consists of three components: mobile devices, a radio access network and a core network. The mobile device might be a smart phone, tablet, or a computer with a USB dongle, but could also be a low-cost sensor with a simple transmitter. The radio access network consists mainly of base stations (mobile phone towers) and is connected to the core network. The base station uses radio waves to relay communications between the mobile device and the core network. The area covered by a base station is called a cell.

Source: [https://theconversation.com/what-is-a-mobile-network-anyway-this-is-5g-boiled-down-102199](https://theconversation.com/what-is-a-mobile-network-anyway-this-is-5g-boiled-down-102199)

---

**Network Redundancy**

Network redundancy is a process through which additional or alternate instances of network devices, equipment and communication mediums are installed within network infrastructure. It is a method for ensuring network availability in case of a network device or path failure and unavailability. As such, it provides a means of network failover.

Source: [https://www.techopedia.com/definition/29305/network-redundancy](https://www.techopedia.com/definition/29305/network-redundancy)
Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behaviour.

Source: https://searchapparchitecture.techtarget.com/definition/object-oriented-programming-OOP

**RSA Algorithm**

Under RSA encryption, messages are encrypted with a code called a public key, which can be shared openly. Due to some distinct mathematical properties of the RSA algorithm, once a message has been encrypted with the public key, it can only be decrypted by another key, known as the private key. Each RSA user has a key pair consisting of their public and private keys.

Source: https://www.comparitech.com/blog/information-security/rsa-encryption/

**Router**

A router connects multiple networks and routes network traffic between them. The router sits in between your Internet connection and your local network.


**SSL Certificate**

SSL stands for Secure Sockets Layer, it's the standard technology for keeping an internet connection secure and safeguarding any sensitive data that is being sent between two systems, preventing criminals from reading and modifying any information transferred, including potential personal details. TLS (Transport Layer Security) is just an updated, more secure, version of SSL.


**Timeliness**

Timeliness refers to the time expectation for accessibility and availability of information. Timeliness can be measured as the time between when information is expected and when it is readily available for use.

Source: https://www.sciencedirect.com/topics/computer-science/timeliness
Transmission Control Protocol (TCP)

The Transmission Control Protocol (TCP) is a transport protocol that is used on top of IP to ensure reliable transmission of packets.


Universal Naming convention (UNC)

UNC is a filename format that is used to specify the location of files, folders, and resources on a local-area network (LAN). The UNC address of a file may look something like this: \\server-name\directory\filename

UNC can also be used to identify peripheral devices shared on the network, including scanners and printers.

Source: https://techterms.com/definition/unc

Uniform Resource Locator (URL)

URL is defined as the global address of documents and other resources on the World Wide Web. The first part of the URL is called a protocol identifier and it indicates what protocol to use, and the second part is called a resource name and it specifies the IP address or the domain name where the resource is located. The protocol identifier and the resource name are separated by a colon and two forward slashes.

Source: https://www.webopedia.com/TERM/U/URL.html