U301 Software development: programming

U301 Rubric

Key knowledge

▪ **emerging trends in programming using artificial intelligence, including:**

- using prompts to generate code
- automated debugging and testing of modules
- code optimisation Googleslides
- responsible and ethical use of artificial intelligence tools

· **characteristics of functional and non-functional requirements, constraints and scope** Google slide the worksheet answers

· **design tools for representing modules, including**:Google slides

- data dictionaries
- mock-ups
- object descriptions
- input-process-output (IPO) chart
https://www.101computing.net/flowchart-prediction-tables/
- pseudocode Pseudocode Worksheet Algorithm vs Pseudocode

· **characteristics of data types, including:**Google slides

- text (character, string)
- numeric (integer, floating point, date/time)
- Boolean

· **characteristics of data structures, including:**Google slides Google slides
textbook page

- one-dimensional arrays

- two-dimensional arrays Google slides
- records (varying data types, field index)

· **characteristics of data sources (plain text (TXT), delimited (CSV) and XML files), including:**Google slides Google slides Summarynotes

- structure
- reasons for use

· **principles of OOP, including:**

- abstraction
- encapsulation Procedural vs OOP
- generalisation
- inheritance

· **features of a programming language, including:**

- local and global variables, and constants
- data types
- instructions and control structures (sequence, selection, iteration/repetition)Worksheet Q&A
- arithmetic, logical and conditional operators
- graphical user interfaces (GUIs)
- functions and methods Worksheet
- classes and objects

· **purposes and features of naming conventions for solution elements (variables, interface controls, code structures), including:**Google slides

- Hungarian notation
- camel casing
- snake casing

· **validation techniques for data, including:**Exam question answers

- existence checking
- type checking

- range checking

· **purposes of internal documentation, including:**

- explaining and justifying data and code structures
- code maintenance
- placeholder comments for future development (stubs)

· **algorithms for sorting and searching, including:**

- selection sort Selection sort Selection sort Example
- quick sort Quicksort notes QuickSortQ&A
- binary search SearchQ&A https://www.101computing.net/linear-search-functions/
- linear search Google slides

· types of errors, including:

- syntax
- logic
- runtime (overflow, index out of range, type mismatch, divide by zero)

· debugging and testing techniques for checking modules function correctly, including:https://www.youtube.com/watch?v=b4p-SBjHh28&t=22s

- use of breakpoints
- use of debugging statements
- construction of relevant test data **Trace table**
- test cases comparing expected and actual output in testing tables.

Google slides textbook page 58-65 Worksheet Q&A

Key skills

▪ interpret solution requirements and designs

· use a range of data types, data structures and data sources

· use and justify appropriate features of an OOP language to develop working software modules Worksheet

· develop and apply suitable naming conventions and validation techniques within modules

· document the functioning of modules using internal documentation

· develop and apply suitable debugging and testing techniques using appropriate test data. Google slides Worksheet Q&A

U302 Software development: analysis and design

Key knowledge

· **AC 1: reasons why individuals and organisations undertake software development projects, including:** Textbook page 87

- increasing productivity and efficiency
- reducing costs
- identifying opportunities to address gaps in the market
- meeting organisational objectives or needs

· **AC1: features of a brief that documents a problem, need or opportunity, including:** Textbook page 89-90

- problem/need/opportunity outline
- proposed users
- programming languages to be used
- feasibility

- originality

· **AC1 features of project management using Gantt charts, including:**Google slides Notes

- identification of tasks
- sequencing of tasks
- time allocation
- dependencies
- milestones
- critical path
- monitoring and documenting the progress of projects

· **AC 2 methods for collecting data to determine needs and requirements, including:** Google slides

- interviews
- observations
- surveys
- reports

· **AC 3 characteristics of functional and non-functional requirements**

· **AC 3 constraints that influence solution development, including:**

- economic
- legal
- social
- technical considerations

· **AC 3 characteristics of solution scope, including:**

- version/solution boundaries

· **analytical tools for depicting the relationships between users, data and systems, including:**

- **AC 3** context diagrams (Level 0) with the components of a system, and entities and data flows

- **AC 3** data flow diagrams (Level 1) with the components of processes, entities, data stores and data flows DFD DFD Rules

- **AC 2** use case diagrams with the components of a system boundary, actors, associations, relationships (includes and extends) and use cases Use Case Diagram Use case theory UseCase Diagram Homework Answers-UseCase Diagram Homework

Text book pages 108- 125

· **AC 3 purpose and features of software requirements specifications, including:**Google slides **Software Requirement Specification (SRS)**

- defining requirements

- constraints Google slides

- scope

- user characteristics

- technical environments

- analytical tools depicting existing processes and systems

· **key legal requirements relating to the intellectual property and ownership and privacy of data used, including:**

- Copyright Act 1968 (Cwlth)

- *Privacy Act 1988* (Cwlth) (APP 1, 3, 6, 8, 9, 11)

- Privacy and Data Protection Act 2014 (IPP 1, 2, 4, 5, 7, 9, 10)

· **file management techniques, including:** Google slides

- the use of naming conventions

- version control

- backups (full, incremental, differential)

- security

- disposal

· **ideation techniques and tools for generating design ideas, including:**

VCAADocument

AC4 Textbook references

Google slides

- mood boards
- brainstorming
- mind maps
- sketches
- annotations

· **criteria for evaluating design ideas and the efficiency and effectiveness of solutions** Google slides Google slide

· **design tools for generating solution designs from design ideas, including:**

- data dictionaries
- mock-ups
- object descriptions
- input-process-output (IPO) charts
- pseudocode

· **characteristics of user experience (UX) and how these affect software design, including:** Google slides

- affordance
- interoperability
- security (authentication and data protection)Google slides
- usability

· **design principles that influence the appearance and functionality of the user interface/s of the software solution, including:**

- alignment
- balance

- contrast
- space
- text formatting
- usability
- navigation.

Key skills

· document a problem, need or opportunity using a brief

· create, monitor and modify project plans using softwareAC1 theory

· select and use a range of methods to collect data

· apply analysis tools to determine solution requirements, constraints and scope

· document an analysis as a software requirements specification SRS Sample

· generate design ideas using appropriate ideation techniques and tools

· develop evaluation criteria for design ideas and the efficiency and effectiveness of the software solution

· produce detailed designs using appropriate design principles and tools.AC5

U401 Software development: development and evaluation

Key knowledge

· **characteristics of efficient and effective solutions, including:**

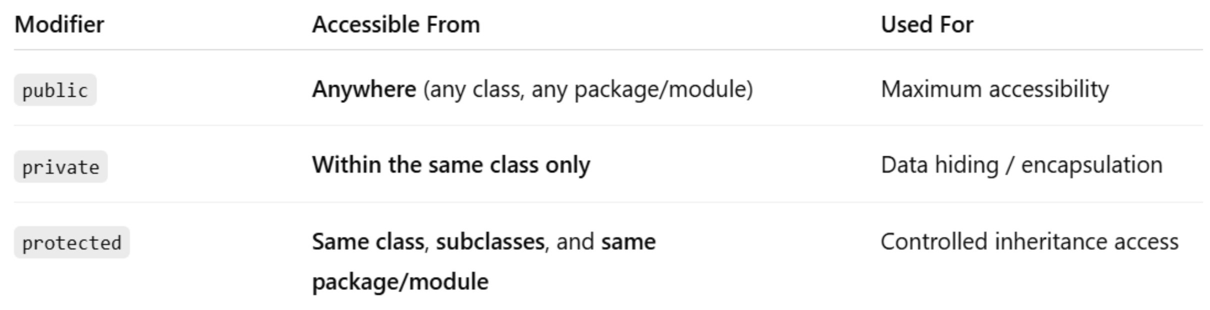
- user-centred design

- clear and concise code
- detailed internal documentation

· **characteristics of data types, data structures and data sources for input, storage and output**

· features of a programming language, including:

- local and global variables and constants
- data types
- instructions and control structures (sequence, selection, iteration/repetition)
- arithmetic, logical and conditional operators
- graphical user interfaces (GUIs)
- functions and methods [Google slides]
- classes and objects
- access modifiers (public, protected and private)



Modifier	Accessible From	Used For
`public`	**Anywhere** (any class, any package/module)	Maximum accessibility
`private`	**Within the same class only**	Data hiding / encapsulation
`protected`	**Same class**, **subclasses**, and **same package/module**	Controlled inheritance access

· **established and innovative approaches to software development, including:**

- the use of code repositories
- application programming interfaces (APIs) and libraries
- artificial intelligence-based (AI) assistants

· **validation techniques, including:**

- existence checking
- type checking

- range checking

· **debugging and alpha testing techniques for checking that solutions meet requirements and function correctly, including the use of**: Google slides

- breakpoints
- commenting out code
- relevant test data
- test cases comparing expected and actual output in testing tables

· **strategies for conducting beta testing, including:**

- construction of a testing plan and test scenarios
- observation of testing scenarios
- documentation of test results

· **features of evaluation strategies, including:**

- evaluation criteria
- time frame
- responsibility

· **techniques for applying evaluation criteria**

· **factors that influence the effectiveness of project plans, including:**

- scope creep
- personnel changes
- technical issues

· **techniques for recording the progress of projects, including:**

- adjustments to tasks
- adjustments to time frames
- annotations to project plans
- monitoring and documenting progress using logs/journals

- **techniques for assessing the effectiveness of a project plan, including:**
  - reviewing the number of changes made to the project plan during the project
  - the reason changes were necessary
  - the impact of changes on the completion of the project.

Key skills

- monitor, modify and annotate project plans as necessary Assessment Criteria 10
- develop a software solution and write internal documentation
- use and apply appropriate data types, data structures and data sources
- develop and apply suitable naming conventions and validation techniques
- select and apply debugging and alpha testing techniques
- prepare and conduct beta testing using appropriate techniques, capture results and recommend modifications to the software solution to address identified issues
- evaluate the efficiency and effectiveness of the software solution Assessment Criteria 9
- assess the effectiveness of the project plan.

Cyber security: secure software development practices

Key knowledge

▪ **goals and objectives of medium and large organisations** Google slides

· **advantages and disadvantages of developing software in-house or externally**

· **types of vulnerabilities and risks within insecure development environments, including**: Google slides

- use of application programming interfaces (APIs)
- malware
- unpatched software
- poor identity and access management practices
- man-in-the-middle attacks
- insider threats
- cyber security incidents
- risks present from software acquired by third parties
- ineffective code review practices
- combined development, testing and production environments

· **security controls used to protect software development practices and data stored within applications, including:**

- version control and code repositories
- robust identity and access management
- encryption
- code review
- regular updates and patches to software
- separated development, testing and production environments

· **threat modelling principles, including:**

- defining security requirements
- identifying and mitigating threats
- confirming threats have been mitigated

· **criteria for evaluating the security of software development practices within an organisation**

· **key legislation and industry frameworks that affect how organisations develop software and control the security and communication of data, including the:**

- Copyright Act 1968 (Cwlth)
- Essential Eight
- Information Security Manual (ISM) (Guidelines for Software Development: Development, testing and production environments; Secure software design and development; Application security testing)
- *Privacy Act 1988* (Cwlth) (APP 1, 6, 8, 9, 11)
- Privacy and Data Protection Act 2014 (IPP 1, 2, 4, 5, 9)

· **ethical issues that arise when developing software, including**: Ethical dilemma from U12

- ineffective security practices
- use of artificial intelligence during development
- intellectual property
- copyright issues

· **mitigation measures to reduce or eliminate threats, vulnerabilities and risks within organisations and development environments**

· strategies for improving the security of software development practices, including:

- onboarding/induction practices and developer training focused on secure development
- development of risk management plans.

Definitions and Risk Management strategies

Key skills

▪ analyse and describe an organisation’s software development practices

· propose and apply criteria to evaluate the effectiveness of the current software development practices

· identify and describe vulnerabilities and risks based on current practices

· identify and discuss the possible legal and ethical consequences to an organisation for ineffective software development practices, and how these could be resolved

· recommend and justify improvements to organisations and their development environments to enhance secure software development practices.