



OTHM LEVEL 5 DIPLOMA IN INFORMATION TECHNOLOGY

Qualification number (RQF): Qualification Reference No. 603/3614/6

Bus504: Software Engineering

Unit code: A/505/9499

ASSIGNMENT BRIEF

The assignment consists of three questions. You are expected to answer all three questions in this assignment. The suggested total word count for the assignment is 2000 words.

Remember, all the words you use to answer the questions, including quotations and citations, count. You should use the mark allocation for each question as a guide to the number of words required as follows:

You must provide a reference list and a word count at the end of your work. The reference list is not included in the word count.

Unit: Software Engineering

Scenario

Ping-source aspires to be an organisation that reflects the globally diverse audience that their products and technology serve. Ping-source believes that in addition to hiring the best talent, a diversity of perspectives, ideas and cultures leads to the creation of better products and services.

Ping-source is and always wants to remain an engineering company. Ping-source hires people with a broad set of technical skills who are ready to take on some of technology's greatest challenges and make an impact on millions, if not billions, of users. At Ping-source, engineers not only revolutionize search, they routinely work on massive scalability and storage solutions, large-scale applications and entirely new platforms for developers around the world.

Ping-source's software engineers develop the next-generation technologies that change how billions of users connect, explore, and interact with information and one another. Ping-source products need to handle information at massive scale and extend well beyond web search.

You have recently joined Ping-source and is given a range of tasks during the probationary one-year period.

Task 1

Task 1 of 3 – Report

Instructions

You need to create a report that will cover the basics of software engineering. You are required to do the following:

1. First part of the report covers modelling languages and their benefits. In the introduction, describe different modelling languages followed by an explanation of the benefits of using modelling languages in system design.

(LO1, AC1.1, 1.2)

2. Second part of the report covers the management of software testing using different strategies. Start the second part with an evaluation of how software testing differs for different strategies.
3. To conclude the report, describe the stages of system testing and evaluate different software testing tools available for the automation of the testing process.

(LO3, AC3.1, 3.2, 3.3)

Delivery and Submission

- 1x Report MS Word processed file (circa 2000 words excluding TOC, diagrams, references and appendices)

Task 2

Task 2 of 3

Design and implement UML

You had a busy first month and the report is now accepted by the management. Now you are asked to demonstrate your software development skills. You are required to demonstrate a series of activities as following:

An automated teller machine (ATM) or the automatic banking machine (ABM) is a banking subsystem (subject) that provides bank customers with access to financial transactions in a public space without the need for a cashier, clerk, or bank teller.

Customer (actor) uses bank ATM to Check Balances of his/her bank accounts, Deposit Funds, Withdraw Cash and/or Transfer Funds. ATM Technician provides Maintenance and Repairs. All these use cases also involve Bank actor whether it is related to customer transactions or to the ATM servicing.

On most bank ATMs, the customer is authenticated by inserting a plastic ATM card and entering a personal identification number (PIN).

(Extension) ATM Technician maintains or repairs Bank ATM. Maintenance use case includes Replenishing ATM with cash, ink or printer paper, Upgrades of hardware, firmware or software, and remote or on-site Diagnostics.

Instructions

- a. Design and build class diagrams using a UML tool for the above user case.
- b. Define/refine class diagrams derived from the given extension code scenario using a UML tool of your choice.
- c. Build an application derived from UML class diagrams using your preferred language.

Delivery and Submission

(LO2, AC2.1, 2.2, 2.3)

3x Lab demonstration files (Lab instructor will sign THREE witness statements for each of the 3 tasks outlining whether each one was successfully demonstrated) *

* Additional files such as pictures during demonstration, video clips on demonstration can be provided on the centre's discretion. (Send All Files on the email)

Task 3

Task 3 of 3 – Lab demonstration

You have successfully completed task 2 and now you are allocated with a final task to complete the probationary period.

Instructions

- a. Develop software using a variety of constructs and apply collections and generics in developing software.
- b. Apply different testing techniques to validate code.

Delivery and Submission

(LO4, AC4.1, 4.2, 4.3)

You are asked to (re)create the popular retro 'Simon' game. In the game 'Four' coloured buttons light up in a specific pattern. After displaying the pattern, the player must repeat the pattern by clicking the buttons in proper order. The pattern gets longer each time the

player completes the pattern. If the player presses a wrong button, the game ends. At the end of the game, the software should display the current score and

also the top 10 scores.

▪ 2x Lab demonstration files (Lab instructor will sign TWO witness statements for each of the 2 activities above outlining whether each one was successfully demonstrated) * (Send All Files on the email)

* Additional files such as pictures during demonstration, video clips on demonstration can be provided on the centre's discretion.

Learning Outcomes and Assessment Criteria

Learning Outcomes- The learner will:	Assessment Criteria- The learner can:
1. Understand modelling languages and their benefits.	1.1 Describe different modelling languages 1.2 Explain the benefit of using modelling languages in system design
2. Be able to design and implement a series of UML class diagrams.	2.1 Design and build class diagrams using a UML tool. 2.2 Define/refine class diagrams derived from a given code scenario using a UML tool. 2.3 Build an application derived from UML class diagrams.
3. Understand the management of software testing using different strategies.	3.1 Evaluate how software testing differs for different strategies. 3.2 Describe the stages of system testing 3.3 Evaluate different software testing tools available for the automation of the testing process.
4. Be able to develop a solution using object-oriented programming.	4.1 Develop software using a variety of constructs. 4.2 Apply collections and generics in developing software. 4.3 Apply different testing techniques to validate code.

Assessment guidance for criterion referenced marking

The assessment grading criteria characterises the level of complexity and demand expected of students at each level of qualification. Please note that these are generic descriptors which apply mainly, though not exclusively, to written academic work.

Any further unit-specific assessment criteria, such as number of words, should be clearly stated in each individual assignment brief.

Result.	Level 3	Level 4	Level 5	Level 6	Level 7
Pass	Detailed answers to all parts of the questions or tasks. Clearly structured and focused, demonstrating overall coherence and in- depth understanding of the unit content and assessment requirements. Evidence of the use of independently sourced material, well applied in all contexts. Very few errors in grammar as appropriate.	Detailed response to all relevant parts of the questions or tasks, with evidence of clear understanding of the issues. Well-structured with evidence of independent reading supporting the argument. Clear evidence of a range of independently sourced material, well applied in all contexts. Very few errors in referencing or grammar or syntax as appropriate.	Very full, independent response to the assignment, applying relevant material well beyond any module input, demonstrating independent study. Excellent understanding and application of relevant theory, concepts and models. Very clear logical structure. Very few errors in referencing or grammar or syntax as appropriate.	Excellent links between relevant ideas, theories and practice. Evidence of independent learning and the ability to engage critically and analytically with a wide range of contextually relevant resource material. Demonstration of original insights supported by well-structured overall argument. Very few errors in referencing or grammar or syntax as appropriate.	The work demonstrates engagement in an academic debate which presents clear evidence of a considered understanding of the topics studied. There is evidence of clear synthesis of theoretical issues and practice. A critical analysis of theoretical and/or applications resulted in originality. Very few errors in referencing or grammar or syntax as appropriate. models practical has

Fail	Little attempt to engage with assignment brief. Learning outcomes not fully met.	Little attempt to engage with assignment brief. Learning outcomes not fully met.	Little attempt to engage with assignment brief. Learning outcomes not fully met.	Little attempt to engage with assignment brief. Learning outcomes not fully met.	<p>Whilst some of the characteristics of a pass have been demonstrated, the work does not address each of the outcomes for the specified assessment task. The work may be an overly descriptive account demonstrating minimal interpretation, and there is very limited evidence of analysis, synthesis or evaluation. No counterarguments or alternative frames of reference are generated or considered.</p>
	Inadequate demonstration of knowledge or understanding of key concepts, theories or practice.	Inadequate demonstration of knowledge or understanding of key concepts, theories or practice.	Inadequate demonstration of knowledge or understanding of key concepts, theories or practice.	Inadequate demonstration of knowledge or understanding of key concepts, theories or practice.	