# **Unit Assessment Pack (UAP) – Cover Sheet**

## **Student and Trainer/Assessor Details**

| **Student ID** |  |
| --- | --- |
| **Student name** |  |
| **Contact number** |  |
| **Email address** |  |
| **Trainer/Assessor name** |  |

## **Course and Unit Details**

| **Course code** |  |
| --- | --- |
| **Course name** |  |
| **Unit code** | ICTPRG501 |
| **Unit name** | Apply advanced object-oriented language skills |

## **Assessment Submission Method**

|  |  |  |
| --- | --- | --- |
| By hand to trainer/assessor | By email to trainer/assessor | Online submission via Learning Management System (LMS) |
| By Australia Post to RTO | Any other method \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Please mention here) | |

**Student Declaration**

|  |
| --- |
| * I certify that the work submitted for this assessment pack is my own. I have clearly referenced any sources used in my submission. I understand that a false declaration is a form of malpractice; * I have kept a copy of this assessment pack and all relevant notes, attachments, and reference material that I used in the production of the assessment pack; * For the purposes of assessment, I give the trainer/assessor of this assessment the permission to:   + Reproduce this assessment and provide a copy to another member of staff; and   + Take steps to authenticate the assessment, including communicating a copy of this assessment to a checking service (which may retain a copy of the assessment on its database for future plagiarism checking).   Student signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Date: \_\_\_\_/\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## **Assessment Plan**

To demonstrate competence in this unit, you must be assessed as satisfactory in each of the following assessment tasks.

| **Evidence recorded** | **Evidence Type/ Method of assessment** | | | **Sufficient evidence recorded/Outcome** |
| --- | --- | --- | --- | --- |
| **Unit Assessment Task 1** | Unit Knowledge Test (UKT) | | | S / NS (First Attempt)  S / NS (Second Attempt) |
| **Unit Assessment Task 2** | Unit Project (UP) | | | S / NS (First Attempt)  S / NS (Second Attempt) |
| **Final result** | C/NYC | **Date assessed** |  | |
| **Trainer/Assessor Signature** |  | |

## **Assessment Conditions**

|  |
| --- |
| **Unit purpose/application** |

This unit describes the skills and knowledge required to undertake advanced programming tasks using an object-oriented programming language.

It applies to individuals who are programmers producing complex object-oriented programming.

No licensing, legislative or certification requirements apply to this unit at the time of publication.

|  |
| --- |
| **What the student can expect to learn by studying this unit of competency** |

* Develop client-server application
* Develop graphical user interface (GUI)
* Build applications
* Debug code
* Test application
* Document system

|  |
| --- |
| **Training and assessment resources required for this unit of competency** |

The student will have access to the following:

* Learner guide
* PowerPoint presentation
* Unit Assessment Pack (UAP)
* Access to other learning materials such as textbooks

The resources required for these assessment tasks also included:

* Access to a computer, the Internet and word-processing system such as MS Word.
* Integrated Development Environment (IDE)
* Help documentation tool
* Simulated assessment environments must simulate the real-life working environment where these skills and knowledge would be performed, with all the relevant equipment and resources of that working environment.

|  |
| --- |
| **Submission instructions** |

Your trainer/assessor will confirm assessment submission details for each assessment task.

|  |
| --- |
| **Academic integrity, plagiarism and collusion** |

**Academic Integrity**

Academic Integrity is about the honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas.

As a student, you are required to:

* undertake studies and research responsibly and with honesty and integrity
* ensure that academic work is in no way falsified
* seek permission to use the work of others, where required
* acknowledge the work of others appropriately
* take reasonable steps to ensure other students cannot copy or misuse your work.

**Plagiarism**

Plagiarism means to take and use another person's ideas and or manner of expressing them and to pass them off as your own by failing to give appropriate acknowledgement. This includes material sourced from the internet, RTO staff, other students, and from published and unpublished work.

Plagiarism occurs when you fail to acknowledge that the ideas or work of others are being used, which includes:

* Paraphrasing and presenting work or ideas without a reference
* Copying work either in whole or in part
* Presenting designs, codes or images as your own work
* Using phrases and passages verbatim without quotation marks or referencing the author or web page
* Reproducing lecture notes without proper acknowledgement.

**Collusion**

Collusion means unauthorised collaboration on assessable work (written, oral or practical) with other people. This occurs when a student presents group work as their own or as the work of someone else.

Collusion may be with another RTO student or with individuals or students external to the RTO. This applies to work assessed by any educational and training body in Australia or overseas.

Collusion occurs when you work without the authorisation of the teaching staff to:

* Work with one or more people to prepare and produce work
* Allow others to copy your work or share your answer to an assessment task
* Allow someone else to write or edit your work (without rto approval)
* Write or edit work for another student
* Offer to complete work or seek payment for completing academic work for other students.

Both collusion and plagiarism can occur in group work. For examples of plagiarism, collusion and academic misconduct in group work please refer to the RTO’s policy on Academic integrity, plagiarism and collusion.

Plagiarism and collusion constitute cheating. Disciplinary action will be taken against students who engage in plagiarism and collusion as outlined in RTO’s policy.

Proven involvement in plagiarism or collusion may be recorded on students’ academic file and could lead to disciplinary action.

|  |
| --- |
| **Other Important unit specific Information** |

N/A

|  |
| --- |
| **Unit outcome** |

* This unit is not graded and the student must complete and submit all requirements for the assessment task for this cluster or unit of competency to be deemed competent.
* Students will receive a 'satisfactorily completed' (S) or 'not yet satisfactorily completed (NS) result for each individual unit assessment task (UAT).
* Final unit result will be recorded as competency achieved/competent (C) or competency not yet achieved/not yet competent (NYC).

|  |
| --- |
| **Prerequisite/s** |

Nil

|  |
| --- |
| **Co-requisite/s** |

Nil

|  |
| --- |
| **Foundation Skills** |

The Foundation Skills describe those required skills (learning, oral communication, reading, writing, numeracy, digital technology and employment skills) that are essential to performance. Foundation skills essential to performance are explicit in the performance criteria of this unit of competency.

|  |
| --- |
| **Relevant Legislation** |

* Australian Human Rights Commission Act 1986
* Age Discrimination Act 2004
* Disability Discrimination Act 1992
* Racial Discrimination Act 1975
* Sex Discrimination Act 1984
* The Privacy Act 1988 (Privacy Act) and Australian Privacy Principles (APPs)
* Occupational Health and Safety Act 2004
* Work Health and Safety Act 2011

|  |
| --- |
| **Principles of assessment and rules of evidence** |

All assessment tasks will ensure that the principles of assessment and rules of evidence are adhered to.

The principles of assessment are that assessment must be valid, fair, flexible, reliable and consistent. The rules of evidence state that evidence must be sufficient, valid, current and authentic.

|  |
| --- |
| **AQF Level** |

AQF levels and the AQF levels criteria are an indication of the relative complexity and/or depth of achievement and the autonomy required to demonstrate that achievement.

All assessment tasks must ensure compliance with the requirements of AQF level and the AQF level criteria. For more information, please visit <http://www.aqf.edu.au/>

|  |
| --- |
| **Further Information** |

For further information about this unit go to <https://training.gov.au/Training/Details/ICTPRG501>

## **Additional Information**

* This information will be managed by the provisions of the Privacy Act and the Freedom of Information Act.)
* Students are required to satisfactorily complete and submit all assessment tasks that contribute to the assessment for a unit.
* Students will be provided with one more attempt to complete this Unit assessment pack (UAP) if trainer/assessor deems them not satisfactorily completed (NS) in any Unit assessment task (UAT).
* Unit Pre-Assessment Checklist (UPAC) will be reviewed by the trainer/assessor to ensure the student is ready for the assessment.
* Feedback regarding this Unit Assessment Pack (UAP) can be emailed to the [compliance](mailto:info@caqa.online) and quality assurance department/administration department in your RTO for continuously improving our assessment and student resources.

## **Feedback to student**

Feedback on students’ assessment performance is a vital element in their learning. Its purpose is to justify to students how their competency was assessed, as well as to identify and reward specific qualities in their work, to recommend aspects needing improvement, and to guide students on what steps to take.

Feedback defines for students what their trainer/assessor thinks is important for a topic or a subject. At its best, feedback should:

* Be provided for each Unit Assessment Task (UAT)
* Guide students to adapt and adjust their learning strategies
* Guide trainers/assessors to adapt and adjust teaching to accommodate students’ learning needs
* Be a pivotal feature of learning and assessment design, not an add-on ritual
* Focus on course and unit learning outcomes
* Guide students to become independent and self-reflective learners and their own critics
* Acknowledge the developmental nature of learning.

*If students have not received proper feedback, they must speak to compliance and quality assurance department/administration department in the RTO/person responsible for looking after the quality and compliance services of the RTO.*

*For more information, please refer to RTO Student Handbook.*

# **Unit Pre-Assessment Checklist (UPAC)**

# **UAT 1 – Unit Knowledge Test (UKT)**

## **Purpose of the checklist**

The pre-assessment checklist helps students determine if they are ready for assessment. The trainer/assessor must review the checklist with the student before the student attempts the assessment task. If any items of the checklist are incomplete or not clear to the student, the trainer/assessor must provide relevant information to the student to ensure they understand the requirements of the assessment task. The student must ensure they are ready for the assessment task before undertaking it.

**Section 1: Information for Students**

* Please make sure you have completed the necessary prior learning before attempting this assessment.
* Please make sure your trainer/assessor clearly explained the assessment process and tasks to be completed.
* Please make sure you understand what evidence is required to be collected and how.
* Please make sure you know your rights and the Complaints and Appeal process.
* Please make sure you discuss any special needs or reasonable adjustments to be considered during the assessment (refer to the Reasonable Adjustments Strategy Matrix and negotiate these with your trainer/assessor).
* Please make sure that you have access to a computer and the internet (if you prefer to type the answers).
* Please ensure that you have all the required resources needed to complete this Unit Assessment Task (UAT).
* Due date of this assessment task is according to your timetable.
* In exceptional (compelling and compassionate) circumstances, an extension to submit an assessment can be granted by the trainer/assessor.
* Evidence of the compelling and compassionate circumstances must be provided together with your request for an extension to submit your assessment work.
* Request for an extension to submit your assessment work must be made before the due date of this assessment task.

## **Section 2: Reasonable adjustments**

* Students with carer responsibilities, cultural or religious obligations, English as an additional language, disability etc. can request for reasonable adjustments.
* Please note, academic standards of the unit/course will not be lowered to accommodate the needs of any student, but there is a requirement to be flexible about the way in which it is delivered or assessed.
* The Disability Standards for Education requires institutions to take reasonable steps to enable the student with a disability to participate in education on the same basis as a student without a disability.
* Trainer/Assessor must complete the section below “Reasonable Adjustment Strategies Matrix” to ensure the explanation and correct strategy have been recorded and implemented.
* Trainer/Assessor must notify the administration/compliance and quality assurance department for any reasonable adjustments made.
* All evidence and supplementary documentation must be submitted with the assessment pack to the administration/compliance and quality assurance department.

|  |  |  |
| --- | --- | --- |
| **Reasonable Adjustment Strategies Matrix (Trainer/Assessor to complete)** | | |
| **Category** | **Possible Issue** | **Reasonable Adjustment Strategy**  **(select as applicable)** |
| 🞎 LLN | 🞎 Speaking  🞎 Reading  🞎 Writing  🞎 Confidence | 🞎 Verbal assessment  🞎 Presentations  🞎 Demonstration of a skill  🞎 Use of diagrams  🞎 Use of supporting documents such as wordlists |
| 🞎 Non-English Speaking Background | 🞎 Speaking  🞎 Reading  🞎 Writing  🞎 Cultural background  🞎 Confidence | 🞎 Discuss with the student and supervisor (if applicable) whether language, literacy and numeracy are likely to impact on the assessment process  🞎 Use methods that do not require a higher level of language or literacy than is required to perform the job role  🞎 Use short sentences that do not contain large amounts of information  🞎 Clarify information by rephrasing, confirm understanding  🞎 Read any printed information to the student  🞎 Use graphics, pictures and colour coding instead of, or to support, text  🞎 Offer to write down, or have someone else write, oral responses given by the student  🞎 Ensure that the time available to complete the assessment, while meeting enterprise requirements, takes account of the student’s needs |
| 🞎 Indigenous | 🞎 Knowledge and understanding  🞎 Flexibility  🞎 Services  🞎 Inappropriate training and assessment | 🞎 Culturally appropriate training  🞎 Explore understanding of concepts and practical application through oral assessment  🞎 Flexible delivery  🞎 Using group rather than individual assessments  🞎 Assessment through completion of practical tasks in the field after demonstration of skills and knowledge. |
| 🞎 Age | 🞎 Educational background  🞎 Limited study skills | 🞎 Make sure font size is not too small  🞎 Trainer/Assessor should refer to the student’s experience  🞎 Ensure that the time available to complete the assessment takes account of the student’s needs  🞎 Provision of information or course materials in accessible format.  🞎 Changes in teaching practices, e.g. wearing an FM microphone to enable a student to hear lectures  🞎 Supply of specialised equipment or services, e.g. a note-taker for a student who cannot write  🞎 Changes in lecture schedules and arrangements, e.g. relocating classes to an accessible venue  🞎 Changes to course design, e.g. substituting an assessment task  🞎 Modifications to physical environment, e.g. installing lever taps, building ramps, installing a lift |
| 🞎 Educational background | 🞎 Reading  🞎 Writing  🞎 Numeracy  🞎 Limited study skills and/or learning strategies | 🞎 Discuss with the Student previous learning experience  🞎 Ensure learning and assessment methods meet the student’s individual need |
| 🞎 Disability | 🞎 Speaking  🞎 Reading  🞎 Writing  🞎 Numeracy  🞎 Limited study skills and/or learning strategies | 🞎 Identify the issues  🞎 Create a climate of support  🞎 Ensure access to support that the student has agreed to  🞎 Appropriately structure the assessment  🞎 Provide information or course materials in accessible format, e.g. a textbook in braille  🞎 Changes in teaching practices, e.g. wearing an FM microphone to enable a student to hear lectures  🞎 Supply of specialised equipment or services, e.g. a note- taker for a student who cannot write  🞎 Changes in lecture schedules and arrangements, e.g. relocating classes to an accessible venue  🞎 Changes to course design, e.g. substituting an assessment task  🞎 Modifications to physical environment, e.g. installing lever taps, building ramps, installing a lift |
|  |  |  |

| **Explanation of reasonable adjustments strategy used (If required)** |
| --- |
|  |

# **Unit Assessment Task (UAT)**

## **Assessment Task 1 - Unit Knowledge Test (UKT)**

**Assessment type:**

* Written Questions

**Assessment task description:**

* This is the first (1) unit assessment task you have to successfully complete to be deemed competent in this unit of competency.
* The Unit Knowledge Test is comprised of eight (8) written questions
* You must respond to all questions and submit them to your Trainer/Assessor.
* You must answer all questions to the required level, e.g. provide the number of points, to be deemed satisfactory in this task
* You will receive your feedback within two weeks - you will be notified by your Trainer/Assessor when results are available.

**Applicable conditions:**

* All knowledge tests are untimed and are conducted as open book tests (this means you are able to refer to your textbook during the test).
* You must read and respond to all questions.
* You may handwrite/use computers to answer the questions.
* You must complete the task independently.
* No marks or grades are allocated for this assessment task. The outcome of the task will be Satisfactory or Not Satisfactory.
* As you complete this assessment task you are predominately demonstrating your written skills and knowledge to your trainer/assessor.
* The trainer/assessor may ask you relevant questions on this assessment task to ensure that this is your own work.

**Resubmissions and reattempts:**

* Where a student’s answers are deemed not satisfactory after the first attempt, a resubmission attempt will be allowed.
* You must speak to your Trainer/Assessor if you have any difficulty in completing this task and require reasonable adjustments (e.g. can be given as an oral assessment)
* For more information, please refer to your RTO Student Handbook.

**Location:**

* This assessment task may be completed in a learning management system (i.e. Moodle) or independent learning environment.
* Your trainer/assessor will provide you further information regarding the location for completing this assessment task.

**Instructions for answering written questions:**

* Complete a written assessment consisting of a series of questions.
* You will be required to correctly answer all the questions.
* Do not start answering questions without understanding what is required from you. Read the questions carefully and critically analyse them for a few seconds, this will help you to identify what is really needed.
* Your answers must demonstrate an understanding and application of relevant concepts, critical thinking, and good writing skills.
* Be concise to the point and write answers according to the given word-limit to each question and do not provide irrelevant information. Be careful, quantity is not quality.
* Be careful to use non-discriminatory language. The language used should not devalue, demean, or exclude individuals or groups on the basis of attributes such as gender, disability, culture, race, religion, sexual preference or age. Gender inclusive language should be used.
* When you quote, paraphrase, summarise or copy information from the sources you are using to write your answers/research your work, you must always acknowledge the source.

**How your trainer/assessor will assess your work?**

* This assessment task requires the student to answer all the questions.
* Answers must demonstrate the student’s understanding and knowledge of the unit.
* If all assessment tasks are deemed Satisfactory (S), then the unit outcome is Competent (C).
* If at least one of the assessment task is deemed Not Satisfactory (NS), then the unit outcome is Not Yet Competent (NYC).
* Once all assessment tasks allocated to this Unit of Competency have been undertaken, trainer/assessor will complete an Assessment plan to record the unit outcome. The outcome will be either Competent (C) or Not Yet Competent (NYC).
* The “Assessment Plan” is available with the Unit Assessment Pack (UAP) – Cover Sheet.

**Purpose of the assessment task:**

* The purpose of this assessment task is to assess the students’ knowledge required for develop applications using advanced object-oriented language skills

## **Assessment Task 1 - Unit Knowledge Test (UKT)**

**Instructions:**

* This is an individual assessment.

The purpose of this assessment task is to assess the students’ knowledge required for develop applications using advanced object-oriented language skills.

* To make full and satisfactory responses you should consult a range of learning resources, other information such as handouts and textbooks, learners’ resources and slides.
* All questions must be answered in order to gain competency for this assessment.
* You may attach a separate sheet if required.
* You must include the following particulars in the footer section of each page of the attached sheets:
  + Student ID or Student Name
  + Unit ID or Unit Code
  + Course ID or Course Code
  + Trainer and assessor name
  + Page numbers
* You must staple the loose sheets together along with the cover page.
* You must attach the loose sheets chronologically as per the page numbers.
* Correction fluid and tape are not permitted. Please do any corrections by striking through the incorrect words with one or two lines and rewriting the correct words.

Resources required to complete the assessment task:

* Learner guide
* PowerPoint presentation
* Unit Assessment Pack (UAP)
* Access to other learning materials such as textbooks
* Access to a computer, the Internet and word-processing system such as MS Word.

Summarise the clients-server app model. Write your response in 100-200 words.

|  |
| --- |
| ***The Client-server model is a distributed application structure that partitions task or workload between the providers of a resource or service, called servers, and service requesters called clients. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and deliver the data packets requested back to the client. Clients do not share any of their resources. Examples of Client-Server Model are Email, World Wide Web, etc.***   * ***Client: When we talk the word Client, it mean to talk of a person or an organization using a particular service. Similarly in the digital world a Client is a computer (Host) i.e. capable of receiving information or using a particular service from the service providers (Servers).*** * ***Servers: Similarly, when we talk the word Servers, It mean a person or medium that serves something. Similarly in this digital world a Server is a remote computer which provides information (data) or access to particular services.***   ***So, its basically the Client requesting something and the Server serving it as long as its present in the database.***  ***Reference:*** GeeksforGeeks. 2020. *Client-Server Model - Geeksforgeeks*. [online] Available at: <https://www.geeksforgeeks.org/client-server-model/> [Accessed 9 November 2020]. |

1. Summarise what a web service is, and the different types that can be implemented by an app. Write your response in 100-200 words.

|  |
| --- |
| ***Web Service is a collection of standards or protocols for exchanging information between two devices or application. There are mainly two types of web services:***   1. ***SOAP Web Services***   ***SOAP stands for Simple Object Access Protocol. It is a XML-based protocol for accessing web services. SOAP is a W3C recommendation for communication between two applications.***  ***SOAP is XML based protocol. It is platform independent and language independent. By using SOAP, you will be able to interact with other programming language applications.***   1. ***RESTful Web Services***   ***Restful Web Services is a lightweight, maintainable, and scalable service that is built on the REST architecture. Restful Web Service, expose API from your application in a secure, uniform, stateless manner to the calling client. The calling client can perform predefined operations using the Restful service. The underlying protocol for REST is HTTP. REST stands for REpresentational State Transfer.***  ***Reference:*** Guru99.com. 2020. *Restful Web Services Tutorial With REST API Example*. [online] Available at: <https://www.guru99.com/restful-web-services.html> [Accessed 12 November 2020]. |

1. Summarise what an applications process is, and how threads can help with the performance of the application. Write your response in 100-150 words.

|  |
| --- |
| ***When an application component starts and the application does not have any other components running, the Android system starts a new Linux process for the application with a single thread of execution. By default, all components of the same application run in the same process and thread (called the "main" thread). If an application component starts and there already exists a process for that application (because another component from the application exists), then the component is started within that process and uses the same thread of execution. However, you can arrange for different components in your application to run in separate processes, and you can create additional threads for any process.***  ***By default, all components of the same application run in the same process and most applications should not change this. However, if you find that you need to control which process a certain component belongs to, you can do so in the manifest file.***  ***When an application is launched, the system creates a thread of execution for the application, called "main." This thread is very important because it is in charge of dispatching events to the appropriate user interface widgets, including drawing events. It is also almost always the thread in which your application interacts with components from the Android UI toolkit (components from the android.widget and android.view packages). As such, the main thread is also sometimes called the UI thread. However, under special circumstances, an app's main thread might not be its UI thread; for more information, see Thread annotations.***  ***The system does not create a separate thread for each instance of a component. All components that run in the same process are instantiated in the UI thread, and system calls to each component are dispatched from that thread. Consequently, methods that respond to system callbacks (such as onKeyDown() to report user actions or a lifecycle callback method) always run in the UI thread of the process.***  ***Reference:*** Android Developers. 2020. *Processes And Threads Overview  |  Android Developers*. [online] Available at: <https://developer.android.com/guide/components/processes-and-threads> [Accessed 12 November 2020]. |

1. Provide an overview of the stages in a Waterfall model, and why this model is suited for large size application development. Write your response in 100-200 words.

|  |
| --- |
| ***The waterfall model is a classical model used in system development life cycle to create a system with a linear and sequential approach. It is termed as waterfall because the model develops systematically from one phase to another in a downward fashion. This model is divided into different phases and the output of one phase is used as the input of the next phase. Every phase has to be completed before the next phase starts and there is no overlapping of the phases. The sequential phases described in the Waterfall model are:***   1. ***Requirement Gathering: All possible requirements are captured in product requirement documents.*** 2. ***Analysis Read: the requirement and based on analysis define the schemas, models and business rules.*** 3. ***System Design: Based on analysis design the software architecture.*** 4. ***Implementation: Development of the software in the small units with functional testing.*** 5. ***Integration and Testing: Integrating of each unit developed in previous phase and post integration test the entire system for any faults.*** 6. ***Deployment: Make the product live on production environment after all functional and nonfunctional testing completed.*** 7. ***Maintenance: Fixing issues and release new version with the issue patches as required.***   ***Reference:*** The Economic Times. 2020. *What Is Waterfall Model? Definition Of Waterfall Model, Waterfall Model Meaning - The Economic Times*. [online] Available at: <https://economictimes.indiatimes.com/definition/waterfall-model> [Accessed 12 November 2020]. |

1. Identify and outline testing techniques suitable for distributed applications. Write your response in 100-200 words.

|  |
| --- |
| ***Distributed application development means developing applications which are executed, or the data used by them is located, on several systems. Usually it is client/server programs.***  ***When testing client/server programs, we must test both the client and server part, and also the communication between them.***  ***Usually, the client part of the application is the front end, the UI. Functionality tests should be performed on it, for example have all input fields proper input validation, is the proper information sent to the server, and so on. Also, UAT (User Acceptance Tests ) should be performed on the front end. Is the user interface user friendly, is it usable for the end user?***  ***The client communicates with the server, usually with sending and receiving responses to HTTP requests, or by calling some API used by the application. This communication should be tested also. Here we may test what happens if we send to the server user input not validated by the UI (we can send it directly with the API or HTTP request ). How will the server react, will it assume that all validation is done by the client (incorrect), or will the server perform its own input validation? (Correct). Performance testing can be done on this communication, how much load can the server handle safely, and if the load more than that, how will it handle it? Will it crash and/or become unresponsive, or will it display, for example, a polite message telling us to try later, while treating previous queued requests?***  ***Functional tests can be done also on the server side of the application, or the backend. Here probably a database will be involved, so performance tests can be run on it too, and also database tests. Of course, security testing should be performed to be as sure as possible that the application is not vulnerable to malicious attacks.***  ***Depending on the requirements of your software, many other tests can be performed, but this is a basic sketch of the needed testing techniques.***  ***Reference:*** 2020. [online] Available at: <https://www.quora.com/What-are-the-testing-techniques-which-are-applied-to-distributed-application-development> [Accessed 12 November 2020]. |

1. Explain what a nested class is and why you might use them. Write your response in 100-150 words.

|  |
| --- |
| ***In Java nested classes are classes that are defined inside another class.***  ***The purpose of a nested class is to clearly group the nested class with its surrounding class, signaling that these two classes are to be used together. Or perhaps that the nested class is only to be used from inside its enclosing (owning) class.***  ***Java developers often refer to nested classes as inner classes, but inner classes (non-static nested classes) are only one out of several different types of nested classes in Java.***  ***In Java nested classes are considered members of their enclosing class. Thus, a nested class can be declared public, package (no access modifier), protected and private (see access modifiers for more info). Therefore nested classes in Java can also be inherited by subclasses as explained in my tutorial about Java inheritance.***  ***Reference:*** Tutorials.jenkov.com. 2020. [online] Available at: <http://tutorials.jenkov.com/java/nested-classes.html> [Accessed 12 November 2020]. |

1. Explain how drag and drop can be implemented between two GUI components in Java or C#. Write your response in 100-150 words.

|  |
| --- |
| ***In case we wish to transfer data from one UI component (Where applicable) to another, one way to carry it out is cut+ paste or copy+ paste. In this case data first gets stored in the clipboard and then from the clipboard it gets transferred over to the other program.***  ***Drag and drop facility is the other way to carry out transfer of information between two UI components (Where applicable). In this case clipboard is not used. In java swing based applications we can implement drag and drop feature to be used.***  ***Drag and drop can be done within different UI components of the same application as well as it can happen between two different applications.***  ***Reference:*** Examples Java Code Geeks. 2020. *Java Swing Drag And Drop Example | Examples Java Code Geeks - 2020*. [online] Available at: <https://examples.javacodegeeks.com/desktop-java/swing/java-swing-drag-drop-example/> [Accessed 12 November 2020]. |

1. Explain how to code 2D graphics in either Java or C#. Write your response in 100-200 words.

|  |
| --- |
| ***The Java library includes a simple package for drawing 2D graphics, called java.awt. AWT stands for “Abstract Window Toolkit”.***  ***There are several ways to create graphics in Java; the simplest way is to use java.awt.Canvas and java.awt.Graphics. A Canvas is a blank rectangular area of the screen onto which the application can draw. The Graphics class provides basic drawing methods such as drawLine, drawRect, and drawString. Here is an example program that draws a circle using the fillOval method:***  import java.awt.Canvas;  import java.awt.Graphics;  import javax.swing.JFrame;  public class Drawing extends Canvas {  public static void main(String[] args) {  JFrame frame = new JFrame("My Drawing");  Canvas canvas = new Drawing();  canvas.setSize(400, 400);  frame.add(canvas);  frame.pack();  frame.setVisible(true);  }  public void paint(Graphics g){  g.fillOval(100, 100, 200, 200);  }  }  ***The Drawing class extends Canvas, so it has all the methods provided by Canvas, including setSize. You can read about the other methods in the documentation, which you can find by doing a web search for “Java Canvas”.***  ***Reference:*** Books.trinket.io. 2020. *Java 2D Graphics | Think Java | Trinket*. [online] Available at: <https://books.trinket.io/thinkjava/appendix-b.html> [Accessed 12 November 2020]. |

# **Unit Assessment Result Sheet (UARS)**

## **Assessment Task 1 – Unit Knowledge Test (UKT)**

## **Student and Trainer/Assessor Details**

|  |  |
| --- | --- |
| **Unit code** | ICTPRG501 |
| **Unit name** | Apply advanced object-oriented language skills |
| **Outcome of Unit Assessment Task (UAT)** | |  | | --- | | **First attempt:** |   Outcome (please make sure to tick the correct checkbox):  Satisfactory (S)  or Not Satisfactory (NS)  Date: \_\_\_\_\_\_\_(day)/ \_\_\_\_\_\_\_(month)/ \_\_\_\_\_\_\_\_\_\_\_\_(year)   |  | | --- | | **Second attempt:** |   Outcome (please make sure to tick the correct checkbox):  Satisfactory (S)  or Not Satisfactory (NS)  Date: \_\_\_\_\_\_\_(day)/ \_\_\_\_\_\_\_(month)/ \_\_\_\_\_\_\_\_\_\_\_\_(year) |
| **Feedback to Student** | |  | | --- | | * **First attempt:** |  |  | | --- | | * **Second attempt:** | |
| **Student Declaration** | * I declare that the answers I have provided are my own work. Where I have accessed information from other sources, I have provided references and or links to my sources. * I have kept a copy of all relevant notes and reference material that I used as part of my submission. * I have provided references for all sources where the information is not my own. I understand the consequences of falsifying documentation and plagiarism. I understand how the assessment is structured. I accept that all work I submit must be verifiable as my own. * I understand that if I disagree with the assessment outcome, I can appeal the assessment process, and either re-submit additional evidence undertake gap training and or have my submission re-assessed. * All appeal options have been explained to me. |
| **Student Signature** |  |
| **Date** |  |
| **Trainer/Assessor Name** |  |
| **Trainer/Assessor Declaration** | I hold:  🗹 Vocational competencies at least to the level being delivered  🗹 Current relevant industry skills  🗹 Current knowledge and skills in VET, *and undertake*  🗹 Ongoing professional development in VET  *I declare that I have conducted an assessment of this candidate’s submission. The assessment tasks were deemed current, sufficient, valid and reliable. I declare that I have conducted a fair, valid, reliable, and flexible assessment. I have provided feedback to the above-named candidate.* |
| **Trainer/Assessor Signature** |  |
| **Date** |  |
| **Office Use Only** | Outcome of Assessment has been entered onto the Student Management System on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (insert date)  by (insert Name) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Unit Pre-Assessment Checklist (UPAC)**

# **UAT 2 – Unit Project (UP)**

## **Purpose of the checklist**

The pre-assessment checklist helps students determine if they are ready for assessment. The trainer/assessor must review the checklist with the student before the student attempts the assessment task. If any items of the checklist are incomplete or not clear to the student, the trainer/assessor must provide relevant information to the student to ensure they understand the requirements of the assessment task. The student must ensure they are ready for the assessment task before undertaking it.**Section 1: Information for Students**

* Please make sure you have completed the necessary prior learning before attempting this assessment.
* Please make sure your trainer/assessor clearly explained the assessment process and tasks to be completed.
* Please make sure you understand what evidence is required to be collected and how.
* Please make sure you know your rights and the Complaints and Appeal process.
* Please make sure you discuss any special needs or reasonable adjustments to be considered during the assessment (refer to the Reasonable Adjustments Strategy Matrix and negotiate these with your trainer/assessor).
* Please make sure that you have access to a computer and the internet (if you prefer to type the answers).
* Please ensure that you have all the required resources needed to complete this Unit Assessment Task (UAT).
* Due date of this assessment task is according to your timetable.
* In exceptional (compelling and compassionate) circumstances, an extension to submit an assessment can be granted by the trainer/assessor.
* Evidence of the compelling and compassionate circumstances must be provided together with your request for an extension to submit your assessment work.
* Request for an extension to submit your assessment work must be made before the due date of this assessment task.

## **Section 2: Reasonable adjustments**

* Students with carer responsibilities, cultural or religious obligations, English as an additional language, disability etc. can request for reasonable adjustments.
* Please note, academic standards of the unit/course will not be lowered to accommodate the needs of any student, but there is a requirement to be flexible about the way in which it is delivered or assessed.
* The Disability Standards for Education requires institutions to take reasonable steps to enable the student with a disability to participate in education on the same basis as a student without a disability.
* Trainer/Assessor must complete the section below “Reasonable Adjustment Strategies Matrix” to ensure the explanation and correct strategy have been recorded and implemented.
* Trainer/Assessor must notify the administration/compliance and quality assurance department for any reasonable adjustments made.
* All evidence and supplementary documentation must be submitted with the assessment pack to the administration/compliance and quality assurance department.

|  |  |  |
| --- | --- | --- |
| **Reasonable Adjustment Strategies Matrix (Trainer/Assessor to complete)** | | |
| **Category** | **Possible Issue** | **Reasonable Adjustment Strategy**  **(select as applicable)** |
| 🞎 LLN | 🞎 Speaking  🞎 Reading  🞎 Writing  🞎 Confidence | 🞎 Verbal assessment  🞎 Presentations  🞎 Demonstration of a skill  🞎 Use of diagrams  🞎 Use of supporting documents such as wordlists |
| 🞎 Non-English Speaking Background | 🞎 Speaking  🞎 Reading  🞎 Writing  🞎 Cultural background  🞎 Confidence | 🞎 Discuss with the student and supervisor (if applicable) whether language, literacy and numeracy are likely to impact on the assessment process  🞎 Use methods that do not require a higher level of language or literacy than is required to perform the job role  🞎 Use short sentences that do not contain large amounts of information  🞎 Clarify information by rephrasing, confirm understanding  🞎 Read any printed information to the student  🞎 Use graphics, pictures and colour coding instead of, or to support, text  🞎 Offer to write down, or have someone else write, oral responses given by the student  🞎 Ensure that the time available to complete the assessment, while meeting enterprise requirements, takes account of the student’s needs |
| 🞎 Indigenous | 🞎 Knowledge and understanding  🞎 Flexibility  🞎 Services  🞎 Inappropriate training and assessment | 🞎 Culturally appropriate training  🞎 Explore understanding of concepts and practical application through oral assessment  🞎 Flexible delivery  🞎 Using group rather than individual assessments  🞎 Assessment through completion of practical tasks in the field after demonstration of skills and knowledge. |
| 🞎 Age | 🞎 Educational background  🞎 Limited study skills | 🞎 Make sure font size is not too small  🞎 Trainer/Assessor should refer to the student’s experience  🞎 Ensure that the time available to complete the assessment takes account of the student’s needs  🞎 Provision of information or course materials in accessible format.  🞎 Changes in teaching practices, e.g. wearing an FM microphone to enable a student to hear lectures  🞎 Supply of specialised equipment or services, e.g. a note-taker for a student who cannot write  🞎 Changes in lecture schedules and arrangements, e.g. relocating classes to an accessible venue  🞎 Changes to course design, e.g. substituting an assessment task  🞎 Modifications to physical environment, e.g. installing lever taps, building ramps, installing a lift |
| 🞎 Educational background | 🞎 Reading  🞎 Writing  🞎 Numeracy  🞎 Limited study skills and/or learning strategies | 🞎 Discuss with the Student previous learning experience  🞎 Ensure learning and assessment methods meet the student’s individual need |
| 🞎 Disability | 🞎 Speaking  🞎 Reading  🞎 Writing  🞎 Numeracy  🞎 Limited study skills and/or learning strategies | 🞎 Identify the issues  🞎 Create a climate of support  🞎 Ensure access to support that the student has agreed to  🞎 Appropriately structure the assessment  🞎 Provision of information or course materials in accessible format, e.g. a text book in braille  🞎 Changes in teaching practices, e.g. wearing an FM microphone to enable a student to hear lectures  🞎 Supply of specialised equipment or services, e.g. a note taker for a student who cannot write  🞎 Changes in lecture schedules and arrangements, e.g. relocating classes to an accessible venue  🞎 Changes to course design, e.g. substituting an assessment task  🞎 Modifications to physical environment, e.g. installing lever taps, building ramps, installing a lift |

| **Explanation of reasonable adjustments strategy used (If required)** |
| --- |
|  |

# **Unit Assessment Task (UAT)**

## **Assessment Task 2 – Unit Project (UP)**

**Assessment type:**

Unit Project (UP)

**Assessment task description:**

* This is the second (2) assessment task you have to successfully complete to be deemed competent in this unit of competency.
* This assessment task requires you to complete a project
* Student is required to evaluate current strategy against the effects of an ICT strategic change, and then develop action plans to implement the change.
* You will receive your feedback within two weeks - you will be notified by your trainer/assessor when results are available.
* You must attempt all activities of the project for your trainer/assessor to assess your competency in this assessment task.

**Applicable conditions:**

* All two activities are untimed.
* You must read and respond to all criteria of the project.
* You may handwrite/use computers to answer the criteria of the project.
* You must complete the task independently.
* No marks or grades are allocated for this assessment task. The outcome of the task will be Satisfactory or Not Satisfactory.
* As you complete this assessment task you are predominately demonstrating your practical skills, techniques and knowledge to your trainer/assessor.
* The trainer/assessor may ask you relevant questions on this assessment task to ensure that this is your own work.

**Resubmissions and reattempts:**

* Where a student’s answers are deemed not satisfactory after the first attempt, a resubmission attempt will be allowed.
* You must speak to your Trainer/Assessor if you have any difficulty in completing this task and require reasonable adjustments (e.g. can be given as an oral assessment).
* For more information, please refer to your RTO Student Handbook.

**Location:**

* This assessment task may be completed in an independent learning environment or learning management system.
* Your trainer/assessor will provide you further information regarding the location of completing this assessment task.

**General Instructions for attempting the project:**

* You will be analyse a scenario and then identify, gather, and analyse data for a business need.
* You will be required to correctly attempt all activities of this assessment task.

## You must concise to the point and write answers according to the given word-limit to each question and do not provide irrelevant information.

## You must use non-discriminatory language. The language used should not devalue, demean, or exclude individuals or groups on the basis of attributes such as gender, disability, culture, race, religion, sexual preference or age. Gender inclusive language should be used.

**How your trainer/assessor will assess your work?**

* This assessment task requires the student to successfully complete and submit a project.
* Answers must demonstrate the student’s understanding and skills of the unit.
* You will be assessed according to the provided performance checklist/ performance criteria.
* Assessment objectives/ measurable learning outcome(s) are attached as performance checklist/ performance criteria with this assessment task to ensure that you have successfully completed and submitted the assessment task.
* If all assessment tasks are deemed Satisfactory (S), then the unit outcome is Competent (C).
* If at least one of the assessment task is deemed Not Satisfactory (NS), then the unit outcome is Not Yet Competent (NYC).
* Once all assessment tasks allocated to this Unit of Competency have been undertaken, trainer/assessor will complete an Assessment plan to record the unit outcome. The outcome will be either Competent (C) or Not Yet Competent (NYC).
* The “Assessment Plan” is available with the Unit Assessment Pack (UAP) – Cover Sheet.

## **Assessment Task 2 - Unit Project (UP)**

**Instructions to complete this assessment task**:

* Please write responses where applicable using a word processor.
* You must include the following particulars in the footer section of each page of the attached sheets:
  + Student ID or Student Name
  + Unit ID or Unit Code
  + Course ID or Course Code
  + Trainer and assessor name
  + Page numbers
* You must staple the loose sheets together along with the cover page.
* You must attach the loose sheets chronologically as per the page numbers.
* Correction fluid and tape are not permitted. Please do any corrections by striking through the incorrect words with one or two lines and rewriting the correct words.
* This submission must be well presented and follow the guidelines and instructions provided.
* Please follow the format as indicated in the template section below.
* One of the most important steps that you can take: proofread your answers and code.
* All RTO policies are in effect, including the plagiarism policy.

**Scenario:**

You have recently been hired as a junior programmer for a large development company. Your immediate manager wants to get a handle on your expertise before assigning you to commercial development tasks. They have designed two small application tests where you can highlight your expertise in the following areas:

* Client-server application functionality
* Using nested classes
* Implementing inheritance
* Implement 2D graphics
* Using multi-threading
* Using drag and drop operations
* Undertaking debugging
* Providing help documentation in GUI format
* Adhering to coding conventions

You will be showcasing these skills in the following activities.

Where appropriate you should adhere to the organisation coding conventions as follows:

**Coding conventions**

Your code should adhere to the following organisational coding convention standards:

* Apps must be built with the existing architectural framework of the organisation
* When naming variables and methods use the camel case convention
* When naming classes use the pascal case convention
* When naming GUI fields/controls you should use descriptive names
* Ensure that all variables, methods, and classes have descriptive names
* Ensure that each method has an internal document header describing what the method does, and what the method parameters are used for if they have been specified
* Code should be indented for ease of readability
* Open and close braces should be used for all code blocks even those that have a single statement
* Ensure that you use if – else conditional statements instead of the ternary operator shortcut
* You should use constants instead of major numbers
* Constants should be named using all capital letters

**Activity 1: (Develop client-server application)**

In this activity you need to develop a GUI client-server application using an object-oriented programming language such as C# or Java, which uses a local client connected to a server based application using the Transmission Control Protocol (TCP).

The design of the client-server application is a simple chat program which allows a client-based app to connect to and send messages to the server-based app. A connection to a database table is required to gather available users that can be used to send messages. The server-based app can send messages to any connected clients-based apps.

This activity is split into several tasks:

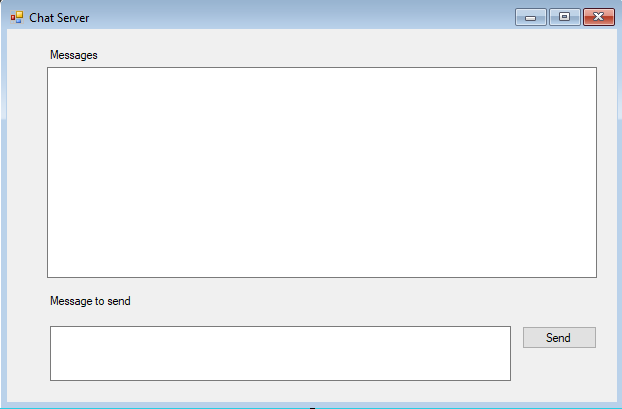
* Develop client-server app and database
* Integrate help documentation
* Debug and test client-server app

**Task 1: Develop client-server app and database**

The client-server app must meet the following requirements.

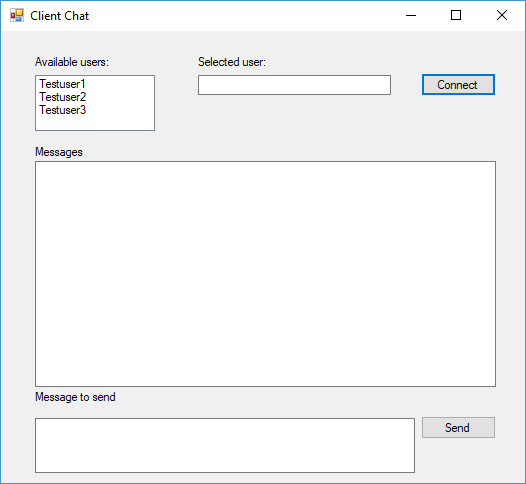
**Functionality requirements**

The design of the Server app should appear similar to that shown below:



It is to contain a message area where messages from clients are received and displayed. It also has an area where a user of this app can type out a message which will be sent to all connected clients when the **Send** button is clicked.

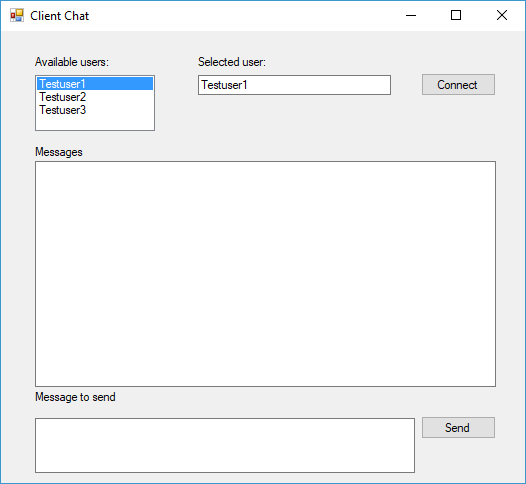
The design of the Client app should appear similar to the following:



A list of available users should be retrieved from a database table which stores all available users. In this instance three test users are available.

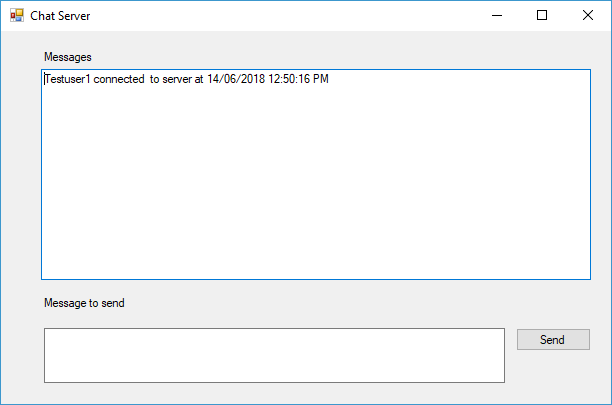
It is to contain a message area where messages from the server are received and displayed. It also has an area where a user of this app can type out a message which will be sent to the server when the **Send** button is clicked.

To connect to the server you must drag and drop a user from the available users’ list box to the selected user text field. An example is shown below where Testuser1 has been dragged from the available users to the selected user text field:

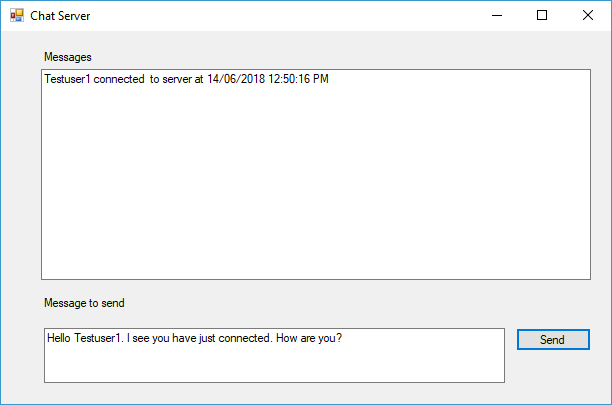


The **Connect** button is used to connect to the server based app using the username that the user has selected. Note that the server must be running and listening for clients otherwise the client will not be able to connect.

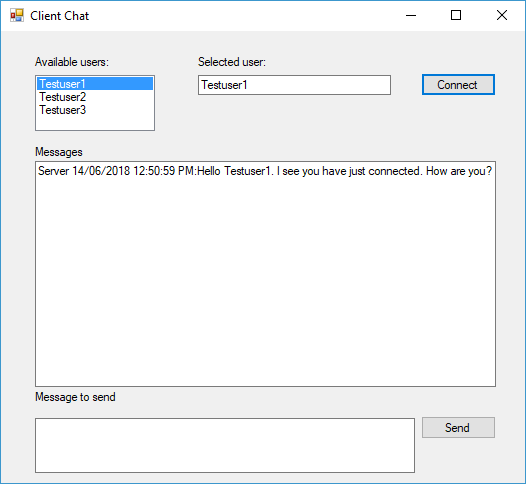
When the **Connect** button is clicked, a TCP connection should be opened to the server app displaying who that the user is connected to the server with a timestamp of the time that the connection was made in the message area of the server. An example is shown below:



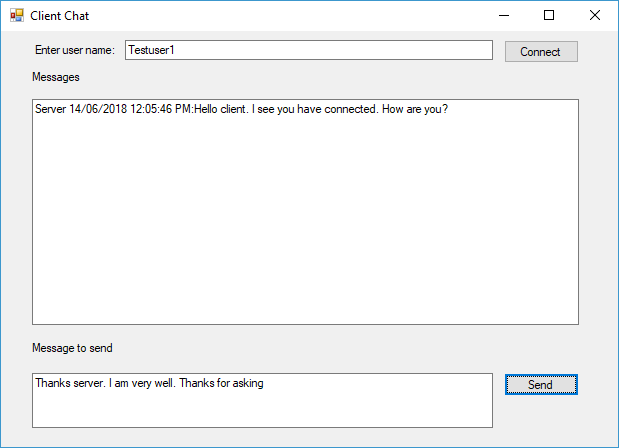
If the user of the server app enters a message and click the **Send** button similar to the example should below:



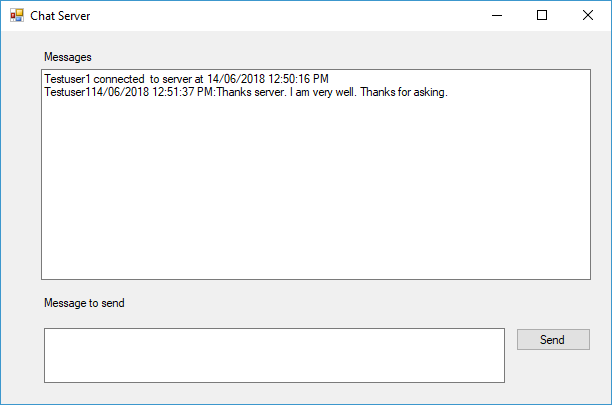
Then the specified message should be transferred to the client app and displayed in the message area:



If the user of the client app enters a message and click the **Send** button similar to the example should below:



Then the specified message should be transferred to the client and displayed in the message area:



You are required to create a database for the above given users and connect with the client-server application.

The database tablefor the available users shouldbe a simple database with a single field called **UserID** which should have a string format of 20 characters. The list of available users should be data bound to the **UserID** field, displaying the available users.

**Coding requirements**

The following coding requirements required for the apps must be included:

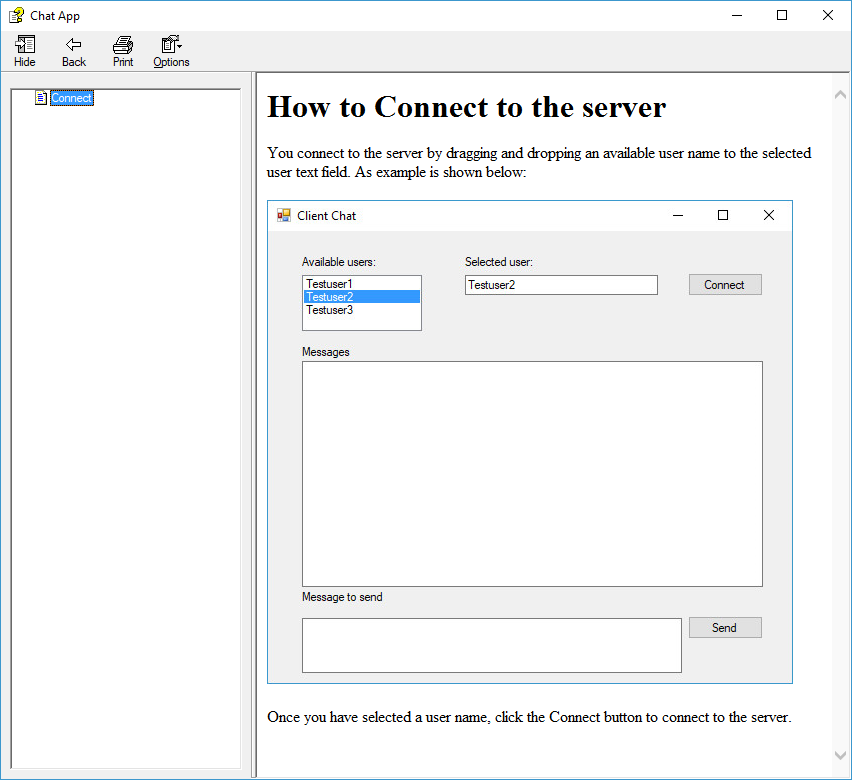
* Coding conventions as given in the scenario must be adhered to.
* The timestamp for the messages must be implemented by using a nested class.
* Must use TCP for communication
* Both apps must implement threading to listen for incoming messages.

You should refer to the online help documentation on design patterns used by the object-oriented language if you require help in coding the apps.

**Task 2: integrate help file**

You need to develop a basic GUI help file to support the user of the client based app. You need to develop an HTML file that will be displayed in a GUI window showing the user how they can connect to the server based app from the client.

The help file should provide a simple instruction in HTML format that tells the user how they can connect. It should contain textual instructions and a single screenshot. An example is shown below of a simple help file that has been created using HTML Help Workshop which has been opened from the client based app:



**Task 3: Testing requirements**

You are to develop a functional testing document using a word processor to test the functions of the apps using the following organisational testing document template

**Functional Test Document**

App to test: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tester: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date Testing Completed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| **Function to test** | **Passed/Fail** |
|  |  |
|  |  |
|  |  |

You need to list every function that the server and client based apps need to undertake.

Once you have documented the functional tests, then you need to undertake testing using the document as a guide, confirming that the function is successful (or not). If there is an issue with any of the functions you should use the debugging tools within the IDE you are using to determine why the function is not working. You should use appropriate debugging tools to trace through the code to determine logic or coding errors, and undertake remedial actions as required until all functions are working as expected.

**Performance criteria checklist for unit assessment task:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Trainer/ Assessor to complete** | | | |
| **Assessment activities to be completed** | * Develop client-server application * For a full project outline, please refer to the student assessment instructions | | |
| **Resources required for the unit assessment task** | * Unit assessment guide template * Integrated Development Environment (IDE) * Help documentation tool * Word processor | | |
| **Does the candidate meet the following criteria** | **Yes** | **No** | **Trainer/Assessor Comments** |
| Developed the client-server app to include drag and drop functionality |  |  |  |
| Developed the client-server app to use the TCP to send and receive messages (RPC) that are based on multiple inheritance |  |  |  |
| Developed the client-server app to use nested classes |  |  |  |
| Developed the client-server app to use multi-threading |  |  |  |
| Adhered to organisational coding standards |  |  |  |
| Integrated a GUI based help file into the client-server app |  |  |  |
| Developed and implemented functional testing and used debugging tools to resolve testing issues |  |  |  |

**Activity 2: (Develop multi-threading 2D app)**

**Note: This activity is in continuation of the previous activity**

In this activity you need to develop a GUI application that implements 2D drawing in a GUI form using multi-threading,

This activity is split into a number of tasks:

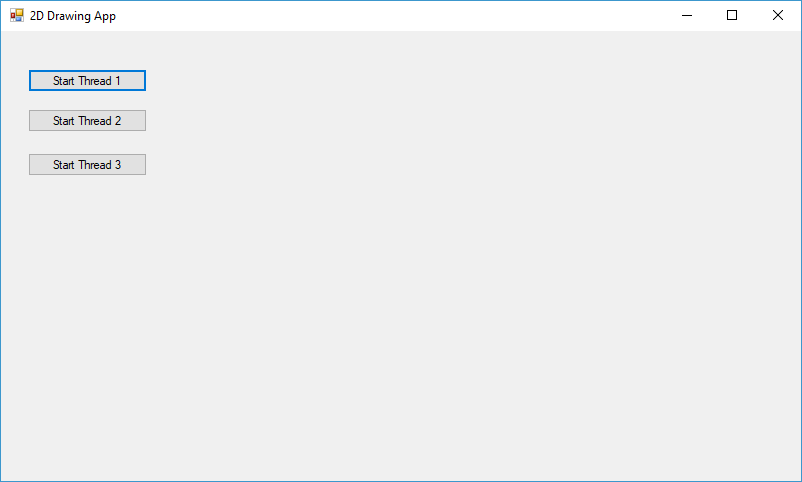
* Develop multi-threading 2D drawing app
* Integrate help documentation
* Debug and test the 2D drawing app

**Task 1: Develop multi-threading 2D drawing app**

The app must meet the following requirements.

**Functionality requirements**

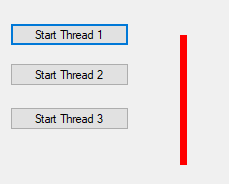
The design of the app should appear similar to that shown below:



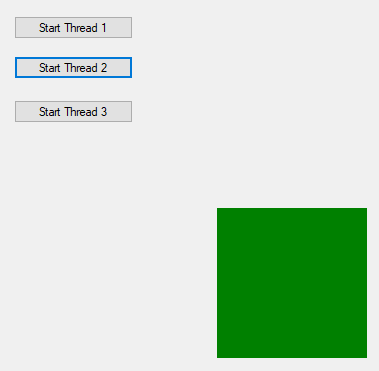
Each button should start a method via a thread that implements drawing a 2D object as follows:

* **DrawObject1**: draws a vertical red line.
* **DrawObject2**: draws a solid green square
* **DrawObject3**: draws a graded blue-green circle

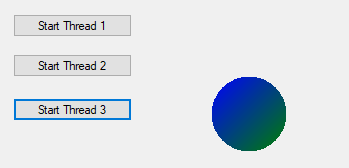
When the **Start Thread 1** button is clicked the **DrawObject1** method should be started and draw a line as follows:



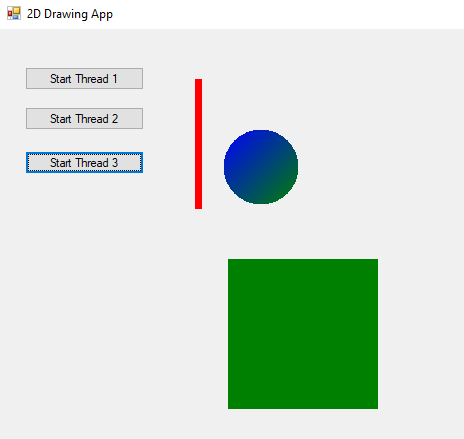
When the **Start Thread 2** button is clicked the **DrawObject2** method should be started and draw a square as follows:



When the **Start Thread 3** button is clicked the **DrawObject3** method should be started and draw a circle as follows:



An example is shown below of all three threads being run concurrently:



**Coding requirements**

The following coding requirements required for the apps must be included:

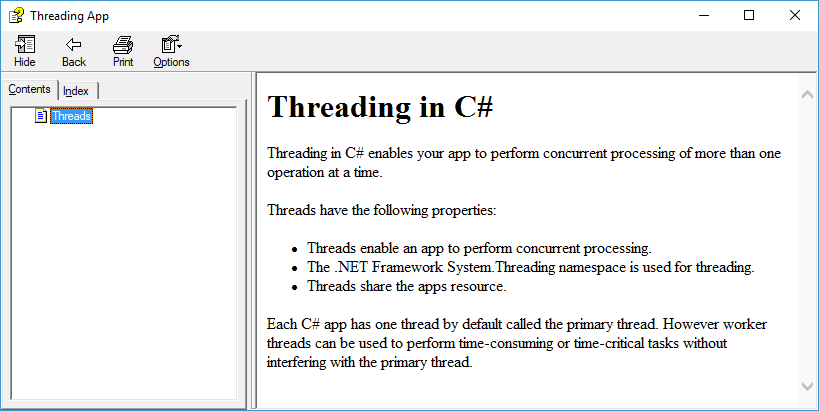
* Coding conventions as given in the scenario must be adhered to.
* The app must implement threading to draw the 2D objects

You should refer to the online help documentation on design patterns used by the object-oriented language if you require help in coding the apps.

**Task 2: integrate help file**

You need to develop a basic GUI help file to support the developers which provides an overview of how thread work. You need to develop an HTML file that will be displayed in a GUI window providing the user an overview of threads.

The help file should provide a simple instruction in HTML format that tells the user what a thread is and how they can be implemented in the language you have created the app in. An example is shown below of a simple help file that has been created using HTML Help Workshop which has been opened from the app:



**Task 3: Testing requirements**

You are to develop a functional testing document using a word processor to test the functions of the app using the following organisational testing document template

**Functional Test Document**

App to test: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tester: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date Testing Completed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| **Function to test** | **Passed/Fail** |
|  |  |
|  |  |
|  |  |

You need to list every function that the app needs to undertake.

Once you have documented the functional tests, then you need to undertake testing using the document as a guide, confirming that the function is successful (or not). If there is an issue with any of the functions you should use the debugging tools within the IDE you are using to determine why the function is not working. You should use appropriate debugging tools to trace through the code to determine logic or coding errors, and undertake remedial actions as required until all functions are working as expected.

**Performance criteria checklist for unit assessment task:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Trainer/ Assessor to complete** | | | |
| **Assessment activities to be completed** | * Develop multi-threading 2D app * For a full project outline, please refer to the student assessment instructions | | |
| **Resources required for the unit assessment task** | * Unit assessment guide template * Integrated Development Environment (IDE) * Help documentation tool * Word processor | | |
| **Does the candidate meet the following criteria** | **Yes** | **No** | **Trainer/Assessor Comments** |
| Developed the app to implement 2D graphic drawing |  |  |  |
| Developed the app to use multi-threading |  |  |  |
| Adhered to organisational coding standards |  |  |  |
| Integrated a GUI based help file into the app |  |  |  |
| Developed and implemented functional testing and used debugging tools to resolve testing issues |  |  |  |

# **Unit Assessment Result Sheet (UARS)**

## **Assessment Task 2 – Unit Project**

## **Student and Trainer/Assessor Details**

|  |  |
| --- | --- |
| **Unit code** | ICTPRG501 |
| **Unit name** | Apply advanced object-oriented language skills |
| **Outcome of Unit Assessment Task (UAT)** | |  | | --- | | **First attempt:** |   Outcome (please make sure to tick the correct checkbox):  Satisfactory (S)  or Not Satisfactory (NS)  Date: \_\_\_\_\_\_\_(day)/ \_\_\_\_\_\_\_(month)/ \_\_\_\_\_\_\_\_\_\_\_\_(year)   |  | | --- | | **Second attempt:** |   Outcome (please make sure to tick the correct checkbox):  Satisfactory (S)  or Not Satisfactory (NS)  Date: \_\_\_\_\_\_\_(day)/ \_\_\_\_\_\_\_(month)/ \_\_\_\_\_\_\_\_\_\_\_\_(year) |
| **Feedback to Student** | |  | | --- | | * **First attempt:** |  |  | | --- | | * **Second attempt:** | |
| **Student Declaration** | * I declare that the answers I have provided are my own work. Where I have accessed information from other sources, I have provided references and or links to my sources. * I have kept a copy of all relevant notes and reference material that I used as part of my submission. * I have provided references for all sources where the information is not my own. I understand the consequences of falsifying documentation and plagiarism. I understand how the assessment is structured. I accept that all work I submit must be verifiable as my own. * I understand that if I disagree with the assessment outcome, I can appeal the assessment process, and either re-submit additional evidence undertake gap training and or have my submission re-assessed. * All appeal options have been explained to me. |
| **Student Signature** |  |
| **Date** |  |
| **Trainer/Assessor Name** |  |
| **Trainer/Assessor Declaration** | I hold:  🗹 Vocational competencies at least to the level being delivered  🗹 Current relevant industry skills  🗹 Current knowledge and skills in VET, *and undertake*  🗹 Ongoing professional development in VET  *I declare that I have conducted an assessment of this candidate’s submission. The assessment tasks were deemed current, sufficient, valid and reliable. I declare that I have conducted a fair, valid, reliable, and flexible assessment. I have provided feedback to the above-named candidate.* |
| **Trainer/Assessor Signature** |  |
| **Date** |  |
| **Office Use Only** | Outcome of Assessment has been entered onto the Student Management System on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (insert date)  by (insert Name) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |