|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Student Name** | Max Crossley saiduasdiuasd | | **Student Number** | | 471075482 |
| **Unit Code/s & Name/s** | Semester 1 - ICTICT418, ICTICT509, ICTDBS502, ICTICT517, ICTPRG418, ICTPRG527, ICTPRG503  Semester 2 - ICTPRG520, ICTPRG502, ICTPRG529, ICTPRG523, ICTGAM528, ICTPRG604, ICTPRG504, ICTPRG532, ICTWEB503 | | | | |
| **Assessment Type** | Case Study  Assignment  Project  Other *(specify)* | | | | |
| **Assessment Name** | Research and Written Submission | **Assessment Task No.** | | | 1 |
| **Assessment Due Date** | Weekly – weeks: 2 to 10  **Semester 1:**  ICTICT418 – Week 2  ICTICT509 – Week 3  ICTDBS502 – Week 4  ICTICT517 – Week 5  ICTPRG418 – Week 6  ICTPRG527 – Week 7  ICTPRG503 – Week 8  **Semester 2:**  ICTPRG520 – Week 2  ICTPRG502 – Week 3  ICTPRG529 – Week 4  ICTPRG523 – Week 5  ICTGAM528 – Week 6  ICTPRG604 – Week 7  ICTPRG504 – Week 8  ICTPRG532 – Week 9  ICTWEB503 – Week 10 | **Date submitted** | | |  |
| **Assessor Name** | Mark O’Reilly | | | | |
| **Student Declaration:** I declare that this assessment is my own work. Any ideas and comments made by other people have been acknowledged as references. I understand that if this statement is found to be false, it will be regarded as misconduct and will be subject to disciplinary action as outlined in the TAFE Queensland Student Rules. I understand that by emailing or submitting this assessment electronically, I agree to this Declaration in lieu of a written signature. | | | | | |
| **Student Signature** |  | | | **Date** |  |
| **PRIVACY DISCLAIMER:** TAFE Queensland is collecting your personal information for assessment purposes. The information will only be accessed by authorised employees of TAFE Queensland. Some of this information may be given to the Australian Skills Quality Authority (ASQA) or its successor and/or TAFE Queensland for audit and/or reporting purposes. Your information will not be given to any other person or agency unless you have given us written permission or we are required by law. | | | | | |

|  |  |
| --- | --- |
| **Instructions to Student** | **Learning Support**  Additional support is available to help you achieve your learning goals. Speak to your teacher or a Learning Skills Centre team member if you feel that you may benefit from some extra support. The Institute provides extra support through the Disability Support Unit and the Learning Skills Centre.  RPL (Recognition of Prior Learning) is available for this unit. Speak to your teacher/assessor to check if you qualify for RPL.  **Conditions of Assessment**  You will need to complete the learning and undertake all assessments satisfactorily to be deemed competent. You are responsible for complying with all assessment item instructions; submission and collection requirements; undertaking assessment tasks honestly and retaining a copy of all assessment items.  You must submit assessment items by the **due date**, unless an extension has been granted by your teacher. Failure to submit assessment items by the due date will result in a “did not submit” being recorded and depending on your circumstances, you may be granted one final resubmission.  To be judged competent in this assessment item the student is required to demonstrate competence in all indicators shown in the marking guide.  **The Classroom as a Simulated Work Environment**  Students must be aware and take responsibility for the problems of working in a shared IT environment. Problems such as noise levels, production flow, interruptions and time variances are common to workplaces. In the simulated environment provided in the classroom these problems can take the form of:   * Other students who continually ask questions or talk aloud while thinking * Fire drills, projector not working, printers running out of paper or toner cartridge * Miscalculating how much work you can do in one day, missing classes and so on.   Some things are unavoidable and you must devise strategies to overcome them, for example, we cannot stop students from asking questions or entering at exiting the class. Other things are unpredictable (e.g. fire drills). You need to be aware and plan and organise your work allowing some extra time for unavoidable and unpredicted events.  **Assessment Criteria:**  To achieve a satisfactory result, your assessor will be looking for your ability to demonstrate key skills/tasks/knowledge to an acceptable industry standard.  Refer to the marking criteria document for a detailed list of items.  **General Instructions:**   * This is a homework-based assessment * In this assessment you will be assessed on the quality of your consideration of each item. Its emphasis is in the demonstration of your knowledge in each specific area. * Each answer is to be no longer than approximately one page. As a guide between 1 and 3 paragraphs are adequate to explain your answers. Diagrams, dot point answers are acceptable where appropriate. * Please submit this assessment document when finished. |
|  | **Number of Attempts:**   * You will receive up to two (2) attempts at this assessment task. * Should your 1st attempt be unsatisfactory (U), your teacher will provide feedback and discuss the relevant sections / questions with you and will arrange a due date for the submission of your 2nd attempt. * If your 2nd submission is unsatisfactory (U), or you fail to submit a 2nd attempt, you will receive an overall unsatisfactory result for this assessment task. * Only one re-assessment attempt may be granted for each assessment task, with the exception of Apprentices or Trainees who are permitted an additional supplementary assessment. * **For more information, refer to the Student Rules.** |
| **Submission details** (if relevant) | Submit your assessment to the allocated dropbox in **Connect** or to the allocated network folder.  Your teacher will provide all the details for the submission system or network.  Your assignment must be saved with your surname\_student number\_unit/cluster\_AssessmentNumber. For example:  **surname\_1234567890\_KE\_1**  For re-submissions, an “R” must be added to the file name. For example:  **surname\_1234567890\_KE\_1\_R**  The Marking Criteria Sheet must be signed and submitted with your work. |
| **Instructions to Assessor** | To be judged competent in this assessment item the student is required to demonstrate an understanding in all items listed. |
| **Note to Student** | An overview of all Assessment Tasks relevant to this unit is located in the Unit Study Guide. |

**Knowledge Evidence Questions**

**Semester 1**

**ICTICT418 - Contribute to copyright, ethics and privacy in an information technology environment**

1. Discuss codes of ethics pertinent to the computing industry
2. Discuss federal and state or territory legislation and policy relevant to an ICT environment and relating to:
   * access and equity
   * copyright and intellectual property
   * workplace health and safety (WHS)
   * privacy
3. What are organisational communication processes and procedures
4. Discuss organisational requirements for customer service
5. Discuss the security features of server operating systems
6. Outline two sample system security procedures.

**ICTICT509 - Gather data to identify business requirements.**

1. What is a client business domain? How might it be relevant to understanding the needs of the client project?
2. In relation to a web-based software project, outline current industry accepted hardware and software products applicable to users, the web-hosting, and to developers.
3. What are programming standards?
4. List stakeholders and outline their respective roles in a web-based software project.
5. Outline 2 approaches or procedures applicable to gathering requirements for a web-based software project.
6. Describe what communication protocols and/or functionality might be required for a web-based software project.

**ICTDBS502 - Design a database**

1. Outline a series of steps that might be taken in analysing a client’s requirements so as to establish:
   1. A conceptual data model,
   2. A complete entity relationship diagram, and…
   3. Required queries and reports.
2. What are the primary components of a conceptual data model?
3. Explain how data redundancy is identified.
4. What are the primary components of a database management system (DBMS)?
5. Explain encryption and authentication as these apply to database security features.
6. What are (describe / give examples of): data types and data structures.
7. Describe the functions and features of databases.
8. What is a logical data model? Why is it important to database queries, screens and reports?
9. What is an object model. How might it be important to database queries, screens and reports?
10. Explain the term ‘scalability’ as it applies to databases.

**ICTICT517 - Match ICT needs with the strategic direction of the organisation**

1. What are the key sections that would be included within an action plan for a software development project?
2. Organisations generally have a strategic plan outlining goals and objectives for business growth and development. How might you anticipate and plan for technical solutions to assist the organisation with its strategic growth?
3. How would you look to assess alternate options that an organisation might purchase or develop to support their technical requirements?
4. List and describe 5 current technologies that have appeared on the market over the last few years.

**ICTPRG418 - Apply intermediate programming skills in another language**

1. What are dynamic variables
2. Describe in detail how small and medium-size application development projects might differ?
3. For a small single text file application, how might file handling and arrays be handled? *(Provide extended detail, examples, diagrams, code and/or pseudo code to demonstrate your understanding)*.
4. Provide a description or example of a user-defined data structure.
5. Describe a limited range of development methodologies and their application.
6. Outline the principles associated with developing a programming language *(such as Java, C#.Net, Python, etc.)*.

**ICTPRG527 - Apply intermediate object-oriented language skills**

1. Explain data structures. ICTDBS502, Q6
2. Explain small-size and medium-size application development. - ICTPRG418, Q2
3. Describe the following object-oriented programming concepts:
   1. Classes and Objects
   2. Inheritance,
   3. Polymorphism
   4. Method Overloading
   5. Association
   6. Multiple Inheritance
4. Define object-oriented programming language
5. In relation to a program with a graphical user interface (GUI):
   1. Outline the steps you would take in setting up such a program.
   2. Describe what programming you would have to put in place to allow the user to interact with the program.
6. Describe internal and external programming / technical documentation.

**ICTPRG503 - Debug and monitor applications**

1. Name at least two (2) examples of the following: (Note: word limit does not apply)
   1. Logging frameworks
   2. Debugging tools
   3. Profiling tools
2. Give reasons on why you would use the following tools when developing an application:
   1. Profiling Tools
   2. Logging and tracing tools
3. Explain the basic principles of:

* computer hardware, networking and components - ICTICT509, Q2
* database-management systems. - ICTDBS502, Q4
* object-oriented programming. - ICTPRG527, Q3
* open-source development tools - ICTPRG523, Q3

1. Java and C# are among the more popular object-oriented programming languages used today. What makes them different from other non-object-oriented programming language?
2. Using an example, discuss the merits (advantages) of open-source development tools.
3. Describe in detail how small and medium-size application development projects might differ in terms of their respective development processes, required tools and techniques.
4. Outline the software development life cycle (SDLC). – ICTPRG418, Q5

**Knowledge Evidence Questions**

**Semester 2**

**ICTPRG520 - Validate an application design against specifications**

1. Explain the database design and implementation. –   
   Database design has 6 phases

Phase 1. Requirements, Collection and Analysis – Questions for customer/ stakeholders, Research, Site Visits, Interviews, Brainstorming sessions, Preferred BDMS, Checklists, Cost, Services. You must collect as much information as possible about the system and its content to fully understand details and grasp the scope to work with.   
Phase 2. Conceptual Design – Forming a concept of how the database will be designed. This phase is important but is just a concept based on the information gathered in Phase 1. Starting with a data model we will draft a blueprint of how the system will be built. Conceptual models need a high level of detail that shows the entities and their relationships. Creating an entry relation diagram (ERD) to give a visual on the plan of the database will wrap up this phase.   
Phase 3. Choice of DBMS – While choosing a database management system you must keep the client in mind. Asking what BDMS the client wants will be a large part of Phase 3. If the client doesn’t mind you need to consider cost, cloud storage and technical factors. At the end of the day, choose the BDMS that works best for you.   
Phase 4. Logical Design – During the logical design phase you will need to sort the data from Phase 2 into a logical data model. Sorting the data by foreign and primary key and setting attributes for the models’ entities will be Phase 4. Identifying the entities and their relationships within the design keeps the data in the 3 NF (normal forms) Following the 3NF keeps the database functional.   
Phase 5. Physical Design – Converting the logical data model progressing though the designer and using the table attributes collated in Phase 4 we can define column data types. Checking constraints to ensure the data is valid is a must. Entities will become tables and the attributes will become columns. Each column within the tables will be assigned a data type. Apply the constraints according to the business rules and user request and consider using an index to keep things tidy. Creating a data dictionary so it can be used as a guide and future reference. Final submission to client for evaluation.   
Phase 6. Implementation – Defining the project scope and deploying the final product.

1. Draw samples of three (3) UML diagrams and explain how each would assist in a software development project.
2. Outline current software development methodologies –

Waterfall Method: Waterfall methodology is used to develop a project though strict steps of development. Most processes flow from the previous phase of development to keep the project on track. Waterfall is used by people who know what they are required to do and have a good comprehension on how to complete the task. Waterfall is usually used for low risk projects but can also be utilized when writing a complicated project. Flowing down the steps will keep the ball rolling and can allow for a clear outline and steps to achieve a desired outcome.

Agile: Agile methodologies can be used in a large variety of situations within software development. Agile is an iterative development cycle. Agile development has a short work window to create a product that can be evaluated by the customer and further worked upon if there are any problems to change. Teams using agile will be able to make changed quickly however there may be a lot of re-writing the code if there are proposed products constantly changing.   
  
Spiral: Spiral model uses repetition to create versions quickly and test to improve the product with successive versions until satisfied. Spiral model goes through planning, design, build and test phases over and over again.

1. Describe, in detail, the system development life cycle (SDLC) –   
   SDLC is the life cycle process that produces high quality software with low cost and the shortest time. SDLC is a plan to develop, alter and maintain a software system. SDLC is used thought the software developer community because it’s functional and proven to be the best practice within the industry. There are more than one model of SDLC but the most popular are Waterfall, Spiral and Agile models.  
     
   Following the stages of best practices ensures the process is done correctly.  
     
   Stage 1: Identifying current problems- this stage is about collecting input from stakeholders, customers, programmers and industry experts. Figuring out what needs to be done to achieve the result and how to improve the system as a goal.   
     
   Stage 2: Planning- Creating a plan evaluating the risks, cost, resources required and what the sub plans will be. Creating a software requirement specification document and planning functionality.  
     
   Stage 3: Design- Changing the software specifications into a plan phase called Design Specification. Collating and sending the results of the previous stages with the Design Specification to all stakeholders allowing them to review and make suggestions. While in the design phase having a plan for collecting and incorporating stakeholder input in the final document is very important. If this stage falls though it can cost the stakeholders and developers time and money.   
     
   Stage 4: Build- Programmers get to work creating code. Working though each previous stage diligently will give the developers a clear outline of what is required from them.  
     
   Stage 5: Test- Test the project for bugs and deficiencies. Fix all problems until we get the desired result for the program.   
     
   Stage 6: Deploy- Deploying the product to the end users while allowing feedback and taking notes on what adjustments can be make, bugs that may have been missed in Stage 5 or changes desired.  
     
   Stage 7: Maintenance- Maintaining the product once it’s been used. Keeping the source code up to date minimising bugs in the future. Working with change in industry and real world change. Keeping the code up to date for the stakeholders is good practice.   
     
   SDLC is a way to plan a project using guidelines to set goals and stages of development. Clear and concise planning of projects is ultimately the most important stage of a project. A poorly planned software project will most likely be flawed, unreliable and worthless.
2. Explain object-oriented programming (OOP)-   
   Object-oriented programming is a type of programming that uses “objects” that hold data within fields that will perform procedures. OOP programs are made by using objects that interact with one another. Data structure becomes an object that includes both data and function. In addition, programmers can create relationships between one object and another.  
     
   OOP has concepts which it uses to create content. Classes and objects are user defined pieces of information that define information used within code. Inheritance classes give the property from one class to another which will give the packages the same behaviour allowing the package to have a parent, child relationship. Association us the relationship between classes of objects that will let one object preform on its behalf.
3. Describe open-source development tools-  
   Open-source development tools such as Git, Eclipse or IntelliJ are used by the developer community to unite and develop as a collective group. Authors and copyright holders of code can upload appropriate licensed or copyright source code to an open-source website and have members of the community develop pieces of code. Open source development allows the whole community to view, work with and access programs that can improve the quality of products worldwide. Open-source software development helps to diversify perspectives within projects and in some cases save money.
4. Describe three (3) software-testing techniques.  
   Alpha Testing- Releasing an unfinished version of the project expecting software testers to identify issues within the project. Alpha testing is conducted towards the end of the development phase which allows minor design changes and bug fixes as result of the testing. Alpha testing is conducted at the development site  
     
   Beta Testing- Beta testing is the step after Alpha testing which allows actual end users to test the product in a closed test. Beta testing is used to create the real environment the product will be exposed to before the official release date.   
   Beta testing is released to a certain number of end users of different areas which allow the company to receive feedback allowing the company to take any action necessary before releasing the final product to the general public.   
     
   Accessibility Testing- Testing to see if the product is accessible for people with disabilities or not.

**ICTPRG502 - Manage a project using software management tools**

1. List and describe 5 Knowledge Areas of Project Management or explain the basic principles of project management.

•Project initiation.

•Project goal.

•Research and create scope - A scope of a project can be defined in terms of its deliverability, defined functionality and its technical structure.

•Identify stakeholders and obtain approval - Involve them and seek feedback at every step of the project development phases.

•Identify resources and risks - Risks are an undeniable part of any project plan. Risk identification is regarded as one of the most important processes for the project plan.

•Make a schedule - Gantt chart.

•Make a budget - Without a proper budget plan any project can run into uncertainty. Top-down, Bottom-up.

•Monitoring and reporting - Project monitoring demonstrates the performance of the outcome whereas the reporting service portrays the formal presentation of the monitoring service.

•Create a risk management plan - Potential hardware and/or software failure, human error, spam, viruses and malicious attacks need to be identified so that they can be managed in a structured way.

•Create a communication plan - A communications plan delivers the measurable results to an organisation. Communications plan with the stakeholders help to achieve goals successfully.

•Develop a change management plan - A change management plan is essential to prepare for any changes that may occur in the Software Development Life Cycle (SDLC). It is important to identify and assess the impacts of change and support individuals with proper training. It is ideal for the project manager to facilitate any required changes within an organisation.

1. Describe the key features of the software development life cycle (SDLC) –   
     
   SDLC is the life cycle process that produces high quality software with low cost and the shortest time. SDLC is a plan to develop, alter and maintain a software system. SDLC is used thought the software developer community because it’s functional and proven to be the best practice within the industry. There are more than one model of SDLC but the most popular are Waterfall, Spiral and Agile models.  
     
   Following the stages of best practices ensures the process is done correctly.  
     
   Stage 1: Identifying current problems- this stage is about collecting input from stakeholders, customers, programmers and industry experts. Figuring out what needs to be done to achieve the result and how to improve the system as a goal.   
     
   Stage 2: Planning- Creating a plan evaluating the risks, cost, resources required and what the sub plans will be. Creating a software requirement specification document and planning functionality.  
     
   Stage 3: Design- Changing the software specifications into a plan phase called Design Specification. Collating and sending the results of the previous stages with the Design Specification to all stakeholders allowing them to review and make suggestions. While in the design phase having a plan for collecting and incorporating stakeholder input in the final document is very important. If this stage falls though it can cost the stakeholders and developers time and money.   
     
   Stage 4: Build- Programmers get to work creating code. Working though each previous stage diligently will give the developers a clear outline of what is required from them.  
     
   Stage 5: Test- Test the project for bugs and deficiencies. Fix all problems until we get the desired result for the program.   
     
   Stage 6: Deploy- Deploying the product to the end users while allowing feedback and taking notes on what adjustments can be make, bugs that may have been missed in Stage 5 or changes desired.  
     
   Stage 7: Maintenance- Maintaining the product once it’s been used. Keeping the source code up to date minimising bugs in the future. Working with change in industry and real-world change. Keeping the code up to date for the stakeholders is good practice.   
     
   SDLC is a way to plan a project using guidelines to set goals and stages of development. Clear and concise planning of projects is ultimately the most important stage of a project. A poorly planned software project will most likely be flawed, unreliable and worthless.

Outline two (2) reasons why software specifications are important.

* It fully addresses all the customer requirements for a product or system.
* Explains the functionality, external interfaces, performance and attributes within the constraints.

1. Identify the methods and importance of version control.  
     
   Version control is the collection of software tools that assist with modifying source code and keeping a record of each version (or branch) of the programs source code. As some large projects can have multiple people working with the same source code it’s important to save individual or different development branches of the source code to minimise any problems regarding bugs, structure failure, errors or processes.   
     
   Sorting the project into branches within a version control helps to separate each developers progress and accountability by giving them their own development version. Version controls give the users a lot of flexibility with what is saved, reverted and added to the source code. Once the code is finalised the team can merge their appropriate branches and have a fully working project.   
     
   Version control has a lot of benefits but having a complete record of every change and modification of the source code means it will be a lot easier to find issues within ANY stage of development.

**ICTPRG529 - Apply testing techniques for software development**

1. Describe the characteristics of the programming language.   
     
   There are 6 main characteristics of a programming language to keep in mind while designing and using a programming language- Portability – the language must be able to run different operating systems to ensure it is usable and reliable throughout all aspects of use. Having different adaptive elements will ensure the longevity of the program application.  
   Efficiency – Consuming as little processing power as possible is an ideal situation when it comes to computing. An efficient computing program will run efficiently enough without taking too much memory leaving the computer to process as fast as possible.  
   Readability – Programs must be written in a logical structure with user friendly perspectives built throughout. Languages can not be difficult to learn.  
   Generality- Coding languages need to be flexible and useable in more than one program. It must have high coupling and cohesion.   
   Documentation- Keeping detailed documentation within the coding language keeps the users in the loop of what’s happening and what the program is capable in its entirety. The language be easy for users to understand it’s application and use.  
   Reliability- Programs must be reliable! Having poor system design can lead to software failure and loss of work.

1. How does an IPO chart assist in the analysis and design of a software project?  
   IPO charts help to describe and show how a business will process information within their system. IPOs are typically designed before the system to keep the input and output streams clear and concise. IPOs help designers understand what you can get from a process.

Outline software development life cycle (SDLC) methodologies. –   
Waterfall Method: Waterfall methodology is used to develop a project though strict steps of development. Most processes flow from the previous phase of development to keep the project on track. Waterfall is used by people who know what they are required to do and have a good comprehension on how to complete the task. Waterfall is usually used for low risk projects but can also be utilized when writing a complicated project. Flowing down the steps will keep the ball rolling and can allow for a clear outline and steps to achieve a desired outcome.

Agile: Agile methodologies can be used in a large variety of situations within software development. Agile is an iterative development cycle. Agile development has a short work window to create a product that can be evaluated by the customer and further worked upon if there are any problems to change. Teams using agile will be able to make changed quickly however there may be a lot of re-writing the code if there are proposed products constantly changing.   
  
Spiral: Spiral model uses repetition to create versions quickly and test to improve the product with successive versions until satisfied. Spiral model goes through planning, design, build and test phases over and over again.

1. Define system layers such as:
   1. the data network – Is a network system that can transfer information and resource data between devices on the same network. Data networks have access points and switches connecting devices to the network making them networked devices.
   2. Hardware – Is the tangible parts of a computer. Hardware is the physical pieces that are put together to create a computer.
   3. operating system – Operating system is like the basic level brain of the computer allowing a user to access further software and schedule tasks
   4. database management systems – Software programs that manages all types of data. Database management systems can store, retrieve and update data within a computer system.
   5. web servers – It’s the internal control system of a website. The server is a computer that stores web server files and scripts. Within the computer there is software running that give access to web users and the files inside the server.
   6. application servers – App servers are software components that run desktop applications. Within the app servers there are web server connectors, runtime libraries and database connectors. App servers run behind webservers and in front of database management systems.
   7. client deployment – Before deploying any software to clients it’s important to do a Software requirement specification to define what is required of the system before deploying software. Attending meetings and finding what is needed from the software is a good place to start. Software deployment follows a deployment process created to give the clients the resources necessary for the task required by the system. Configuring the target computers before the installation of the package data to reduce data loss and minimise slowing work places down due to lack of resources as the software is being deployed. *(deploying software to the hardware that is to be used by a client).*
2. Explain the processes and techniques related to small-size application development. – While building a small sized application it’s important to plan the project to maximise productivity. Following a guided process of building software products will give the development phase a push in the right direction.  
     
   Analysing the problem and working towards a potential solution while conducting market research will be the first step of any project. While out researching you must gather the requirements of the system and programmers. Holding meetings with clients and stakeholders gives the project manager good scope of requested system requirements and functionality.   
     
   Adapting an agile or spiral methodology for smaller projects is a popular approach but using a methodology that suits the situation will be best. Once the methodology is written out, devising a plan to create the application though diagrams of the system (IPO for businesses or ERDs for databases for example) you need a conceptual model before any coding and building can start.   
     
   Once the planning phase of the software system design has been completed, coding can commence. Keeping within the coding guidelines in Question 1, building the project will be the easy part of the project.  
     
   Upon completing the code the functionality and reliability of the system will need to be tested for bugs and breaks. Testing is an important phase to iron wrinkles that may be present. Although this is a small program testing is still necessary and should be thorough.  
     
   Deploying and maintaining the system will be the final step of software building, once the project is deployed a good programmer will maintain the system to ensure quality of life for the customer.

**ICTPRG523 - Apply advanced programming skills in another language**

1. Identify and describe three (3) programming languages. Indicate which of these provide for the development of GUI interfaces.   
   C# - Is a general purpose programming language that focuses on object orientated language. C# was created by Microsoft and is an easy language to learn with high useability and portability. C# is compatible with GUI designers like Microsoft Visual Studio which makes designing and implementing GUIs quick and easy. C# is an important language that holds high functionality and is used a lot.   
     
   Java Script – Java is another language that uses object orientated and class-based methods. Java has been used in the industry for 23 years and still has a huge number of users writing every day. Java was designed to used as little implementation dependencies between systems as possible. Java is intended to be a write once run anywhere language that will give java the flexibility to run on all platforms that support java without any complication. Java has GUI designers within the library, but I found the spring layout method takes a lot of time with a more trial and error.  
     
   Python – Python is a general use programming language that focuses on code readability. Python is a high level, general purpose programming language using object orientating approach which eases the logic of writing code as well as a large library to assist programmers build their projects. Python has a huge number of GUI programming toolkits available with a huge library incorporated within Tkinter. Tkinter provides an extensive GUI designer running off a powerful object orientated interface.
2. Identify and describe languages for a GUI environment. –
3. Describe in detail how medium-size and large-scale application development projects might differ?  
     
     
   Medium-size – Medium sized projects are approximated to have 6 to 12 month timeframe estimation until completion and have usually 3-5 team members working together. Although it is smaller than a large scale project, medium sized projects must follow the project guidelines laid out during the Initiation phase of the project lifecycle.  
   Medium projects vary in complexity but are usually difficult to solve which gives a small amount of flexibility in the timeframe window. Medium teams have a emphasis on getting things done to the best of their ability and will impact a number of areas once the change developed is implemented. Cost of Medium sized applications can be from 25k to 100k depending on the situation.   
   Medium sized projects must have a beginning, scope and planned processes as well as an end result. Methodologies can be a bit more flexible in medium sized projects as the employer may want more than once change or development thoughout the project life cycle. Adapting a waterfall approach with Agile development in parts will be an efficient way to crunch out parts of the job necessary to the solution.   
   Once a medium project is completed there will be testing and bug fixes wherever necessary to provide a complete project up to the clients standard of expectation.   
     
   Large Scale- Large scale projects are approximated to be completed in more than 12 months with 6+ team members working on the job. Large scale work projects cost a lot of money, time and resources to complete so the budget is a lot higher than small or medium sized projects coming in at over $250,000. Large scale projects can impact a whole system which leads to a high level of risk to the project and its dependencies. Starting a large scale project can be a good way to increase reputation within the community although, there are higher risks.   
   Large scale projects must adhere to more strict guidelines and timelines laid by the project manager, failure to follow steps and methodologies though the whole project may result in colossal failure of the final product and inflated cost to the client and development team time and money. Implementing phases of Initiation, planning, execution and closure keeps the project goal concise and tangible while leaving the team with a benchmark project to work towards.   
   Large scale projects have little flexibility of development steps as there many moving parts to configure. Adapting a waterfall is as a main methodology keeps the project running smoothly. Waterfall is focused on running though a plan step by step and only progressing once a milestone has been completed. Teams must follow the liner design to stay on track.
4. Describe the complex data structures: Linked Lists and Binary Trees.  
   Linked Lists – Linked lists are used to create a sequence of links that will store item data in a linear structure allowing a connection link to the first link. Linked lists are useful for their “list” elements that store data and connects sequences of data structures together. Having the linked lists makes the elements easily inserted or removed without reallocation of a whole structure.   
     
   Binary Trees – Binary trees create a ranking or grouping value to a data structure giving the data structural relationships. Having relationships between data allows efficient insertion, deletion and searching within the flexible data subtrees. Elements within a binary tree are linked similar to a linked list that contains pointers. Binary trees have parent elements that are directly above another element and child elements which are below their parent element. Parent elements can only have 2 children (left and right child).
5. How would you link or connect a third-party supplied library within your software project?
   * + 1. Import the built in Java Types
       2. Add a third-party JAR file to the projects user class path within your java model though the model properties window
       3. Run the code generator within the console window by changing the installations *bin* directory and run the batch *xmog.bat*
       4. Click the toolbar and type *Ctrl – F* to view the file window that allows the Java Type package to be selected.
       5. Find the code generators directory though *doc./tutorial/lib* and select your JAR file
       6. Check the user class path from step 2
       7. Save the model as you wish and open it in a text editor
       8. Check for the import() command which will mean it’s being executed.

**ICTGAM528 - Create games for mobile devices**

1. List and describe two (2) currently successful mobile game devices.  
     
   Nintendo Switch – Nintendo released the switch mobile gaming device in 2017. It allows users to take the console anywhere they will have hands free. The Switch is a gaming console capable of online and offline gaming. Nintendo switch has motion sensing, touch screen, detachable controllers and uses game cards.   
     
   iPhone – A very famous mobile game device capable of gaming with a huge library of games to choose from. Mobile phone gaming has become increasingly popular as mobile phones have quickly developed into amazing machines. Mobile phones are extremely popular with estimated 5.13 billion owning a mobile phone.
2. Summarise three (3) current tools, technologies and programming languages needed for creating applications on mobile devices products.
3. Summarise the technical constraints that mobile devices place on design and development of games.
4. Outline suitable programming languages for mobile devices. – Q2, above
5. Describe how you might estimate the cost of a mobile game development project.  
   Firstly, having an idea on what the project will be and what the final product will look like- without a project scope you can’t create a or manage cost. Creating a meeting with potential stakeholders and clients regarding the scope of the project and resources needed will give an idea of cost to develop. Having a clear idea of what the project will and methodologies involved give those involved an idea of how to budget and estimate cost to create and maintain the mobile game system.
6. Explain the importance of assessing risks within a mobile software development project.
7. List and describe the roles of five (5) different team members that may be required in the process of creating a game for a mobile device.

* Researcher – Researchers will be needed to conduct field work regarding the project scope, budget, requirements and even finding people to work on the task at hand.
* Project Manager – Project manager will be in charge of keeping the project on track, ensuring work is progressing and staying within budget and timeframe allowed
* Graphical designer – Graphical designer will create the models and art work for the game.
* Technical writers – Will be writing documentation and reports for the closure and initiation of the game
* Programmers – Will write the code, fix bugs and maintain the code.

1. Outline two techniques for concept visualisation and development.

**ICTPRG604 - Create cloud computing services**

1. Which of the following are the most appropriate development tools for creating web services that can be deployed on the cloud and describe each tool?

* UML Modelling Tools
* WSDL Generators
* Microsoft Visio
* Netbeans

1. Suggest two other such development tools and indicate how they are applicable.
2. Draw a diagram clearly illustrating how a client connects to a cloud-based application, which in turn connects to a web service. Clearly note and describe the hardware and software (infrastructure) required for each section or aspect of this.
3. List and explain what would need to be considered in applying object-oriented programming to a cloud-based solution.
4. Describe (potentially with one or more supporting diagram(s)) how to access and manipulate the database content on the web using HTML and XML.
5. Define the term “big data” in relation to cloud computing.

**ICTPRG504 - Deploy an application to a production environment**

1. Describe the basic principles of database management systems. - ICTDBS502, Q4
2. Under what circumstances might you use a Waterfall Software Development Life Cycle? Describe the standard phases within a Waterfall SDLC.
3. Define system layers such as: - ICTPRG529, Q4
   1. the data network
   2. hardware
   3. operating system
   4. database management systems
   5. web servers
   6. application servers, and…
   7. client deployment (deploying software to the hardware that is to be used by a client).
4. When writing code or scripts for deploying your application to a production environment, what are the components that you will need to consider in order for it to be ready for the end-users?
5. Why is it important to know the policies, procedures and security protocols of an organisation when deploying software?

**ICTPRG532 - Apply advanced object-oriented language skills**

1. Describe the required technologies for developing web applications.
2. Explain the different approaches to implementing inter-process communication in either Java or C#.Net.
3. What project management and development strategies would you utilise to develop a large-scale application?
4. Explain the different testing techniques that you can use in distributed application development.
5. Describe in detail the steps in implementing a third-party supplied library for performing common programming tasks.

**ICTWEB503 - Create web-based programs**

1. Describe in detail:
   1. Three (3) methods or techniques that would assist you in analysing a client’s web project
   2. Three (3) considerations or approaches that would assist you in designing a client’s web project
2. What are programming control structures and how might they be applicable to a web-based application project?
3. Summarise web programming concepts including:
   1. authentication and web security
   2. hypertext transfer protocol (HTTP)
   3. session management
   4. stateless programming.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | | | **Assessment Task**  **- Written – All Units** |  | |  | |   ASSESSMENT SUMMARY | | |
| **Student Name** | | **Student Number** |
|  | |  |
| **Qualification Code & Name** | | **Stream / Specialisation** |
| ICT50718 - Diploma of Information Technology | | Software Development |
| **Unit/s of Competency Code** | **Unit/s of Competency Name** | |
| **Semester 1** | | |
| ICTICT418 | Contribute to copyright, ethics and privacy in an information technology environment | |
| ICTICT509 | Gather data to identify business requirements | |
| ICTDBS502 | Design a database | |
| ICTICT517 | Match ICT needs with the strategic direction of the organisation | |
| ICTPRG418 | Apply intermediate programming skills in another language | |
| ICTPRG527 | Apply intermediate object-oriented language skills | |
| ICTPRG503 | Debug and monitor applications | |
| **Semester 2** | | |
| ICTPRG520 | Validate an application design against specifications | |
| ICTPRG502 | Manage a project using software management tools | |
| ICTPRG529 | Apply testing techniques for software development | |
| ICTPRG523 | Apply advanced programming skills in another language | |
| ICTPRG532 | Apply advanced object-oriented language skills | |
| ICTWEB503 | Create web-based programs | |
| ICTPRG604 | Create cloud computing services | |
| ICTWEB503 | Deploy an application to a production environment | |
| ICTGAM528 | Create games for mobile devices | |
| **Assessment Task Number** | **Assessment Name** | |
| Various | Assignment | |
| **Student Declaration**  I declare that this assessment is my own work. Any ideas and comments made by other people have been acknowledged as references. I understand that if this statement is found to be false, it will be regarded as misconduct and will be subject to disciplinary action as outlined in the TAFE Queensland Student Rules. I understand that by me emailing this assessment item, I agree to this Declaration in lieu of a written signature. | | |
| **Student Signature** | | **Date** |
|  | | /     /20 |

| MARKING CRITERIA | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Attempt** | **Result** | | **Date** | | | | |
| 1st Attempt | Satisfactory | Unsatisfactory | /     /20 | | | | |
| 2nd Attempt | Satisfactory | Unsatisfactory | /     /20 | | | | |
| Supplementary (apprentice / trainees only) | Satisfactory | Unsatisfactory | /     /20 | | | | |
|  | | | | | | | |
|  | | | | **1st Attempt** | | **2nd Attempt** | |
| **Did the student satisfactorily:** | | | | **S** | **U** | **S** | **U** |

**Semester 1 units**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | Contribute to copyright, ethics and privacy in an information technology environment |  |  |  |  |
| 2 | Gather data to identify business requirements |  |  |  |  |
| 3 | Design a database |  |  |  |  |
| 4 | Match ICT needs with the strategic direction of the organisation |  |  |  |  |
| 5 | Apply intermediate programming skills in another language |  |  |  |  |
| 6 | Apply intermediate object-oriented language skills |  |  |  |  |
| 7 | Debug and monitor applications |  |  |  |  |

**Semester 2 units**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Validate an application design against specifications | | | | |  |  |  |  |
| 2 | Manage a project using software management tools | | | | |  |  |  |  |
| 3 | Apply testing techniques for software development | | | | |  |  |  |  |
| 4 | Apply advanced programming skills in another language | | | | |  |  |  |  |
| 5 | Apply advanced object-oriented language skills | | | | |  |  |  |  |
| 6 | Create web-based programs | | | | |  |  |  |  |
| 7 | Create cloud computing services | | | | |  |  |  |  |
| 8 | Deploy an application to a production environment | | | | |  |  |  |  |
| 9 | Create games for mobile devices | | | | |  |  |  |  |
| **Reasonable Adjustment applied to assessment** *(if ‘yes’ record details)* | | | | | | | | | |
| No | | Yes | Details: |  | | | | | |
| **Assessor Name** | | | | | **Assessor Signature** | | | | |
| TBA | | | | |  | | | | |
| **Summary of Feedback / Action Plan** | | | | | | | | | |
|  | | | | | | | | | |