**Part C**

**Questions 1-12**

**App development and XML written assessment**

**Requirements**

Answer each of the twelve (12) questions. Do not write more than 100 words for each question.

Copy and Paste written assessment for each of the questions below into a new Word Document – or – write your answers directly into this document. Name or rename the document to: StudentNumber\_MDXML\_33131-01.docx. For example, 665437\_MDXML\_33131-03.docx.

|  |
| --- |
| **QUESTION 1: SUMMARISE MOBILE APP CONSTRAINTS**  Provide a brief summary of at least three design constraints that should be taken into account when developing apps for hand-held devices.  Memory Management – Mobile devices (especially older ones) have limited memory so if your application requires a lot to run or it is memory inefficient it will perform poorly on a mobile.  Short Battery Life – Most mobiles have quite short battery life and if your application is very power intensive it can force users to stop using the application as it drains their battery far too fast and in turn makes their phone unusable for its main purpose.  Small Screen Size – Mobiles naturally have smaller screen sizes than a regular computer and therefore consideration needs to be given to the readability and usability of the application on a smaller screen. The app my need a different design completely for smaller displays.  Interruption – Since these applications are on mobile, and the primary purpose of a mobile is to /send and receive phone calls and text messages the applications needs to be able to be interrupted at any point during its use and then be resumes without having adverse effects on the functionality or performance of the application. |
| **QUESTION 2: SUMMARISE OBJECT-ORIENTED PRINCIPLES (OOP)**  Provide a brief summary of what the following OOP are:   * Encapsulation   Encapsulation is simply data hiding. It is the process of making sure that variables of a class are not able to be interreacted with by another class. It then makes use of Get methods which allow the user to read the value and Set methods which allow the user to assign a value to a variable.   * Polymorphism   Polymorphism is the name given when a bunch of subclasses all inherit from the same base class. This allows certain classes to use the same method but in their own way. An example of this is a dog and a cat class which both inherit from the animal class. The animal class might have the method talk. Both the cat and the dog class also have the method talk however when it is called from the cat class it Meows and when it is called from the dog class it Barks.   * Inheritance   Inheritance is when another class inherits all the properties and methods from another class. The class that inherits everything is called the subclass and that it inherits from is called the super class. This can be chained many times for example.  You could have an animal class and there are two subclasses which inherit from animal being mammal and reptiles. Mammal can have another class inherit from it for example Whale. This means that Whale has all the properties and methods of both the Animal and the Mammal class as well as any it might have of its own where as Mammal only has properties and methods from the Animal class. |
| **QUESTION 3: SUMMARISE VISUAL PROTOTYPING**  Describe the user-interface design principles. Include a brief summary of how you would use, wireframes, mockups and prototypes.  All three choices can either be sued separately or in conjunction with one another. Wireframing involves a very quick sometimes hand drawn sketch of the UI and its functionality. It is meant o convey the purpose of the application in the simplest of terms. A mock-up is a more realistic looking design of the applications GUI. It will often look very similar to the end product but at this stage is just normally screenshots of a non-functioning program. A Prototype generally looks almost exactly the same as the finished product and often has some actual functionality. It will most likely already have the code to go between different screens and may even have certain areas of the application full functional for demonstration purposes.  They are all used at different stages of development. A wireframe or mockup can most likely be used at the beginning to demonstrate to the development team what needs to happen and the prototype is often something shown to clients or potential shareholders as they are more impressed with a semi working product. |
| **QUESTION 4: SUMMARISE XML**  Provide a brief explanation of what XML is and what it is primarily used for.  XML is a mark up language similar to HTML. It uses code tags to wrap up information. XML on it owns does not actually do anything. XML is primarily designed to carry data and transfer data between applications. XML is very useful in sharing information in a standardised way throught he use of APIs, also known as web services. |
| **QUESTION 5: SUMMARISE Language integration for web design**  Research and explain how HTML, CSS, JavaScript and XML are used in web design.  HTML is the layout of the webpage, where things go, where they are getting their information and how it all comes together. CSS is how the webpage is styled. While you can do all of the styling on the HTML file itself it is much easier and neater to use a CSS (Cascading Style Sheet) to do all the styling. JavaScript is how things on the website do other things. When you go to a weather website most likely there is some JavaScript being run in the background using some form of API to retrieve that data. JavaScript is the only true programming language out of the 4 mentioned. Another example is filling out forms on a Webpage there is most likely some JavaScript code telling the website how to respond and what to do when the submit button is clicked.  XML is useful when you want to send data in a standardised way to another user. This is also generally how APIs operate. The user sends the request for data and expects it to come back in a certain format that they can manage. |
| **QUESTION 6: SUMMARISE MOBILE APP PLATFORMS**  Provide a brief summary of the following three mobile platforms:   * Android   Android is the most popular platform. The following manufactures all use Android for their mobile devices: Samsung, HTC, LG, Motorola, Sony Xiaomi and Asus. Android is written on a Linux Operating system. Android applications are mostly written in Java. Android apps are mostly developed using the follow IDEs: Android Studio, Eclipse with ADT and Visual Studio using Xamarin.   * iOS   iOS is developed by Apple. It is based on OS X from apple computers. All apple devices use the iOS. iOS apps have previously be primarily written in Objective C and C programming languages however recently the new programming language Swift has taken over and most developers have swapped to Swift now. The main IDEs for iOS development are: XCode and Visual Studio with Xamarin plugin.   * Windows 10 mobile   Windows 10 mobile is developed by Microsoft and based on the windows 10 operating system. It is the third most used behind Android and iOS. The following languages are used for windows 10 development: C#, XAML, HTML5, JavaScript and C++. The main Ide to use when developing a Windows 10 application is Visual Studio. |
| **QUESTION 7: SUMMARISE XML TREE HIERARCHY**  Briefly explain the XML tree and how elements can be related to one another within the tree structure.  The XML tree refers to how elements can either be a Parent of a Child in relation to another element. In this example, you can see that Room is nested inside house. This means that Room is a child of House and furthermore House is a Parent of Room. All properties that House has room will also have however House will not have any properties unique to Room.  <House>  <Room>  </Room>  </House> |
| **QUESTION 8: SUMMARISE ELEMENTS AND ATTRIBUTES**  Provide a brief summary of what an XML element and attribute are.  XML Elements are the containers that can hold other elements, text and attributes. An XML element may also be empty. XML attributes are designed to hold data related to a specific element. In this example <Dog bread=”Husky”> the attribute is husky. An attribute is basically a description of what that specific element is. Whereas an element is everything including the tags all the way to the end tag. |
| **QUESTION 9: SUMMARISE THE World Wide Web**  Research and describe the main features and services of the world wide web (WWW).  The World Wide Web is where documents can be retrieves using a Uniform Resource Locator (URL) via the internet. IT works across all platforms and generally uses a web browser to access the information. Some of the features include access to data from across the world in real time, ability to access banking and pay bill from a browser interface and the ability to connect and communicate with other people from all over the world in an instant. |
| **QUESTION 10: SUMMARISE COMMON XAML ELEMENTS**  Provide a brief overview of the main difference between HTML and XAML.  XAML and HTML are completely different mark-up languages. HTML is the standard language of the internet whereas XAML is derived from XML and is used to describe content presentation for parsers that render WPF and Silverlight. The main difference really is that they are not the same at all. XAML is not used for web pages at all and it is used to store data in a standardised way. |
| **QUESTION 11: Information System feature and source characteristics**  Explain briefly what an AppBar is used for in a Universal Windows app and the main XAML elements required to create the AppBar.  The AppBar represent the container control that holds app UI component for commanding and experiences. It has the generic windows App Bar features on it so it is kept uniform across applications. Not all applications will use this as it is a feature not many will need.  The main XAML element required is:  <AppBar>  Content  </AppBar> |
| **QUESTION 12: Data Modelling**  Explain Data Modelling.  Data Modelling is an easy way to represent or explain how an application will work or flow. It is often the first step in the designing of an OOP. It is the process of making sure that the model you have created will meet all the clients requirements and because it is only a model it can be easily changed without much work, this means that you cut down on redevelopment time if the application needs to be changed as all the changes should be made before the actual application has even started to be coded. Stakeholders are often interested in seeing some data modelling before investing so it is a good idea to have a firm grasp on how to use it effectively to portray your design. |