**Name Heng Wei Yao**

**Date 22/10/21**

**Part 1 PRELIMS .NET TEST ANSWERS**

|  |
| --- |
| 1. Thread refers to parallel work or code execution which enables the user to perform multiple tasks simultaneously. It enables the user to move on to the next line of code before the first line code of completes its execution which enables better efficiency especially for tasks that require a lot of processing. |
| 2. Asynchronous programming is mainly used for time consuming tasks and it allows the task to be completed in the background while the program continues on. Parallel programming is completing multiple sub task in order to fully complete one big main task.  Asynchronous programming    Parallel programming |
| 3. Throwing the exception resets the stack trace which would allow errors to originate from handle exception while throw would preserve the original offender. In the picture below, exception is caught method 1 goes back to original method while method 2 would throw it off and not return. |
| 4. You pass a parameter into the thread by declaring a new thread and passing in the information before starting the thread. |
| 5. Key features of OOP are abstraction, polymorphism, inheritance and Encapsulation |
| 6. The types of SDLC models are Water fall model, Spiral model, Agile model, iterative model |
| 7. The order is based on hierarchy, starting from the variables being initialized child variables will initialize followed by parent variables and base class constructor will initialize with parent first followed by child.    Parent class    Child class |
| 8.By using the base keyword, we can access members from a derived class we can also override methods created from the derived class. |
| 9. Try block will run first attempting to complete the task and if any exception Is caught, it will stop the execution and run the catch block code depending on which exception is caught. Finally, regardless of task completion or exception caught, it will always run.  In the example below it will attempt to parse string to double and if any one of it fails it will move on to catch block to display the error message. If no exception is caught it will move on and parse every single variable. Regardless of exception being caught or no it will always print done. |
| 10. Finally will be ran after the execution if try catch block and regardless of the outcome it will always be executed without fail. In this example, it will always print Done at the end of he execution |
| 11. No we cannot declare delegates in an interface we can however, declare events within an interface. An interface only contains signatures of methods, properties, indexers and events. |