**1) What is an exception?**

-An exception is a problem that arises during the execution of a program. A C# exception is a response to an exceptional circumstance that arises while a program is running.

**2) What happens in a try block if the program executes without errors?**

-If the program executes without any errors, all the statements within the “try” block will be executed sequentially until completion.

**3) How does the catch mechanism work for unhandled exceptions?**

-If there is an unhandled exception, the catch mechanism is called to handle the exception. The catch mechanism is a block of code that is executed if the error occurs in the “try” block.

**4) What happens in a program if an exception block fails to handle a particular error?**

-If the exception block fails to handle a particular error, the method exits and execution returns to the calling method.

**5) What is the parent class for all exceptions? How does this work?**

-In C#, all exceptions are derived from the base class “exception” (which derives from the “object” class). The exception class is divided into two branches called “ApplicationException” and “SystemException”. SystemException is a base class for all program code generated errors and ApplicationException is a base class for all application related exceptions.

**6) How do you determine the type of an error?**

-You can determine the type of an error by using the GetType() method.

**7) What is the purpose of integer checking?**

-The purpose of integer checking is to ensure that the value of an integer is within range of the variable parameters so that it can be handled by the compiler.

**8) What does the finally block do?**

-A “finally” block is group of code that executes when control leaves a “try” block and is always execute, regardless of whether an exception was thrown or not.