1 What is a possible situation that leads to the exception?

At present, there are nine possibilities, which are

a) NullReferenceException: When you try to access a null member of an object, you may encounter a NullReferenceException.

b) IndexOutOfRangeException: An IndexOutOfRangeException can occur when you attempt to access an array element with an index that is outside the array's bounds.

c) StackOverflowException: When a programme recurses too deeply, it may cause a StackOverflowException.

d) OutOfMemoryException: An OutOfMemoryException might occur when a programme attempts to allocate more memory than is available.

e) DivideByZeroException: A DivideByZeroException can occur when a programme attempts to divide a number by zero.

f) ArgumentNullException: An ArgumentNullException may be thrown when a method is called with a null parameter.

g) ArgumentOutOfRangeException: An ArgumentOutOfRangeException can occur when a method is called with an argument that is outside of the valid range.

h) FormatException: A FormatException can occur when a software attempts to convert a string to a number but the string is not in a valid format.

I System exception: A SystemException might occur when a programme attempts to perform an unauthorised operation.

2 Who is in charge of throwing the exception: you as a programmer or the runtime system? In theory, should you throw exceptions of this type? Explain your answer If you need to throw the exception, what details would you provide as a message to the user(caller)?

The exception is handled by the runtime system. You should not throw this exception as a coder. To throw an exception, use the "throw" statement. The Exception object is then highlighted. Each of these exceptions includes a message and a description of the error.. The general types are

a) NullReferenceException:

b) IndexOutOfRangeException:

c) StackOverflowException:

d) OutOfMemoryException:

e) DivideByZeroException:

f) ArgumentNullException:

g) ArgumentOutOfRangeException:

h) FormatException:

i) System exception:

3 What parameters would you include in the message?

a) NullReferenceException: Include the name of the null object in the message.

b) IndexOutOfRangeException: Include the name of the array and the index that is out of range in the message.

c) StackOverflowException: Include the name of the method that caused the stack overflow in the message.

d) OutOfMemoryException: Include the name of the method that caused the out of memory error in the message.

e) DivideByZeroException: Include the name of the method that caused the divide by zero error in the message..

f) ArgumentNullException: Include within the message the name of the method that was called with a null argument.

g) ArgumentOutOfRangeException: Include within the message the name of the method that was called with an out of range parameter.

h) FormatException: Include within the message the name of the method that caused the format problem.

I) System exception: Include in message the name of the method that caused the system issue.4 Can the exception be generally caught (and therefore handled)?

If the exception occurs, should you generally catch this exception type or is it better to pass such exception to the user (caller)? It is not enough to say yes or no; you must provide an argument to support your answer.

In most cases, the exception can be caught. ArgumentNullException: This exception is thrown to determine whether or not the input is null. If the input argument is null, the exception ArgumentNullException is thrown; otherwise, the programme runs properly. b.) Format deviation: Exceptional format

5 Is the exception a case when you want to avoid this exception to occur in your application in general?

If so, would be your action as a programmer to avoid it?

The exception is used when you don't want this exception to occur in your application in general. You might avoid it as a programmer by ensuring that operations do not violate the rules before performing them.