7 Lecture - CS506

Important Subjective

Certainly, here are 10 short-answer questions related to the introduction to exceptions in programming, along with their answers:

Question 1: What is an exception in programming?

Answer: An exception in programming is an unexpected event or error that occurs during the execution of a program and disrupts the normal flow of code execution.

Question 2: How do exceptions help in programming?

Answer: Exceptions help in handling unexpected situations that can cause errors or program crashes. They allow developers to gracefully handle these situations, ensuring that the program can recover or terminate in a controlled manner.

Question 3: What is the purpose of the try-catch block?

Answer: The try-catch block is used to catch and handle exceptions. Code inside the try block is monitored for exceptions, and if any occur, they are caught and processed by the catch block.

Question 4: What happens if an exception is thrown within the try block but there's no corresponding catch block?

Answer: If an exception is thrown within the try block but there's no corresponding catch block to handle it, the program will terminate abruptly, and an error message will be displayed.

Question 5: How can you raise an exception manually in your code?

Answer: You can raise an exception manually using the `throw` keyword followed by an instance of an exception class. This allows you to indicate that a specific exceptional condition has occurred.

Question 6: What is the purpose of the finally block in exception handling?

Answer: The finally block is used to specify code that should be executed regardless of whether an exception is thrown or not. It's typically used for cleanup tasks, such as releasing resources or closing files.

Question 7: What is a stack trace?

Answer: A stack trace is a list of method calls that were in progress when an exception was thrown. It provides information about the sequence of calls leading up to the point where the exception occurred.

Question 8: What is the difference between checked and unchecked exceptions?

Answer: Checked exceptions are exceptions that are checked by the compiler at compile time, and the programmer is required to handle or declare them. Unchecked exceptions, on the other hand, are not checked by the compiler and include runtime exceptions.

Question 9: Can you catch multiple exceptions in a single catch block?

Answer: Yes, you can catch multiple exceptions in a single catch block by specifying multiple exception types separated by pipes (|).

Question 10: Why is proper exception handling important in programming?

Answer: Proper exception handling is important because it improves the robustness and reliability of programs. It allows programs to gracefully handle unexpected errors, prevents crashes, aids in debugging, and provides a better experience for users.