| def add\_bonus(balance):  balance += 100 # Adding a bonus to the local copy  print(f"Balance inside function: {balance}")  # Original balance before calling the function  account\_balance = 1000  add\_bonus(account\_balance)  # Check the balance after the function call  print(f"Original balance outside function: {account\_balance}") |
| --- |

| def add\_extra\_credit(grades):  for i in range(len(grades)):  grades[i] += 5 # Adding extra credit to each grade    # Original list of grades  student\_grades = [80, 85, 90]  add\_extra\_credit(student\_grades)    # Check the grades after the function call  print(f"Updated grades: {student\_grades}") |
| --- |

| # In Python, a function simulating pass by name  def simulate\_pass\_by\_name(expression):  return expression + expression # The expression is re-evaluated    x = 5  result = simulate\_pass\_by\_name(x + 2) # Expression x + 2 will be re-evaluated  print(result) |
| --- |

| FileInputStream file = null;  try {  file = new FileInputStream("file.txt");  // Read the file content  } catch (FileNotFoundException e) {  System.out.println("Error: File not found.");  } finally {  if (file != null) {  try {  file.close(); // Ensuring the file is closed  System.out.println("File has been closed.");  } catch (IOException e) {  System.out.println("Error while closing the file.");  }  }  } |
| --- |

| public class ExceptionPropagationExample {  public static void main(String[] args) {  try {  methodA();  } catch (Exception e) {  System.out.println("Caught exception in main: " + e);  }  }    public static void methodA() throws Exception {  methodB();  }    public static void methodB() throws Exception {  throw new Exception("Something went wrong in methodB");  }  } |
| --- |

| try {  methodA();  } catch (FileNotFoundException e) {  System.out.println("File not found: " + e.getMessage());  } catch (IOException e) {  System.out.println("I/O error: " + e.getMessage());  } catch (Exception e) {  System.out.println("An unexpected error occurred: " + e.getMessage());  } |
| --- |

| public static void methodA() throws Exception {  try {  methodB();  } catch (FileNotFoundException e) {  System.out.println("File not found in methodA");  } catch (Exception e) {  throw new Exception("Exception in methodA", e);  }  } |
| --- |

| public static void methodA() throws Exception {  try {  methodB();  } catch (Exception e) {  System.out.println("Handling exception in methodA");  throw e; // Rethrow the exception to be handled by the caller  }  } |
| --- |

| public static void methodA() throws Exception {  methodB(); // Let methodB throw the exception to the caller  } |
| --- |

| int fd = open("file.txt", O\_RDONLY);  if (fd == -1) {  perror("Error opening file");  // Handle error appropriately  } |
| --- |

| if (fd == -1) {  perror("Failed to open file");  } |
| --- |