## **Error Handling**

Error handling allows you to gracefully handle exceptions that may occur during program execution.

### **try-except Block**

*# Basic try-except*  
**try**:  
 number = int(input("Enter a number: "))  
 result = 10 / number  
 print(f"Result: {result}")  
**except**:  
 print("An error occurred")

### **Catching Specific Exceptions**

**try**:  
 number = int(input("Enter a number: "))  
 result = 10 / number  
 print(f"Result: {result}")  
**except** ValueError:  
 print("Invalid input. Please enter a number.")  
**except** ZeroDivisionError:  
 print("Cannot divide by zero.")

### **try-except-else-finally**

**try**:  
 number = int(input("Enter a number: "))  
 result = 10 / number  
**except** ValueError:  
 print("Invalid input. Please enter a number.")  
**except** ZeroDivisionError:  
 print("Cannot divide by zero.")  
**else**:  
 *# Executed if no exceptions occur*  
 print(f"Result: {result}")  
**finally**:  
 *# Always executed, regardless of whether an exception occurred*  
 print("Calculation attempt completed")

### **Raising Exceptions**

**def** set\_age(age):  
 **if** age < 0:  
 **raise** ValueError("Age cannot be negative")  
 **return** age  
  
**try**:  
 user\_age = set\_age(-5)  
**except** ValueError **as** e:  
 print(e) *# "Age cannot be negative"*

**Exercise 14**: Create a program that asks the user for a filename, then tries to open and read the file, handling exceptions appropriately (file not found, permission denied, etc.).