**Exceptional Handling Practice Questions**

**1. What is the role of try and exception block?**

The role of try and exception block is to hyandle errors that might occur while running the code.

**Try and except:**

The codes that might raise an error is placed under try blog, if exception catches the error it will take the action as specified by user.

**For E.g,**

**try:**

number = int(input("Enter a number: ")) # This may raise a ValueError if input is not a number

print(f"The number you entered is {number}")

**except ValueError:**

print("Error: Please enter a valid integer.")

**2. What is the syntax for a basic try-except block?**

**try:**

number = int(input("Enter a number: ")) [ ***This may raise a ValueError if input is not a number]***

print(f"The number you entered is {number}")

**except ValueError:**

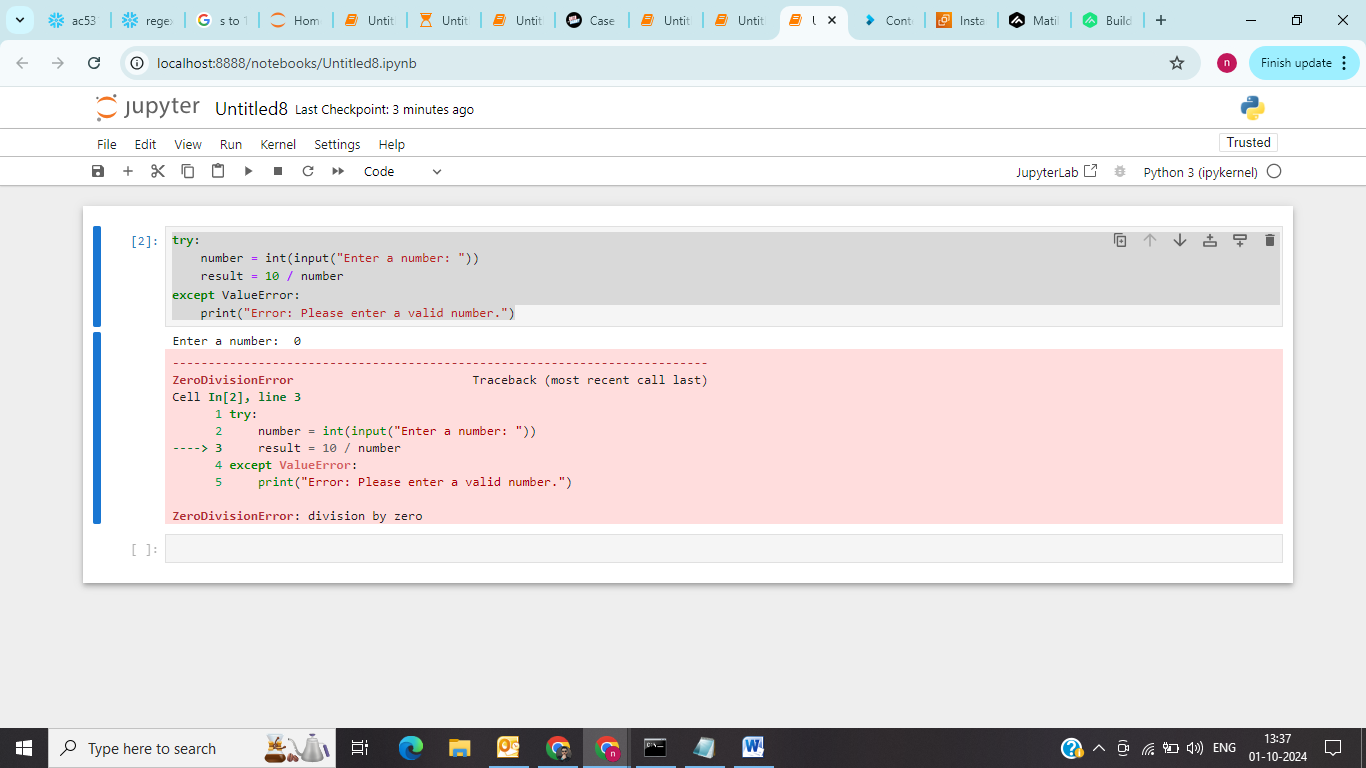
print("Error: Please enter a valid integer.") [**Code to handle error]**

**3. What happens if an exception occurs inside a try block and there is no matching except**

**block?**

The error will get caught and program will not give the expected results. It will give (Traceback (most recent call last)) error.

For e.g:



**4. What is the difference between using a bare except block and specifying a specific exception**

**type?**

|  |  |
| --- | --- |
| **Bare except** | **Specifying exception type** |
| It will catch every possible error that can might occur. | It will catch only specific given type of error. |
| It makes debugging difficult as it doesn’t show the specific type of error. | It makes debugging easy as only the specified error will come. |

**Bare Except E:g**

**try:**

**number = int(input("Enter a number: "))**

**result = 10 / number**

**except:**

**print("An error occurred.")**

**Specifying exception E:g**

**try:**

number = int(input("Enter a number: ")) # This may raise a ValueError if input is not a number

print(f"The number you entered is {number}")

**except ValueError:**

print("Error: Please enter a valid integer.")

**5. Can you have nested try-except blocks in Python? If yes, then give an example.**

**Yes we can have ,**

**try:**

number = int(input("Enter a number: "))

**try:**

result = 10 / number # This could raise a ZeroDivisionError

print(f"The result of 10 divided by {number} is {result}")

**except ZeroDivisionError:**

print("Error: You cannot divide by zero.")

**except ValueError:**

print("Error: Invalid input, please enter a valid integer.")

**6. Can we use multiple exception blocks, if yes then give an example.**

**Yes we can have ,**

**try:**

number = int(input("Enter a number: "))

**try:**

result = 10 / number # This could raise a ZeroDivisionError

print(f"The result of 10 divided by {number} is {result}")

**except ZeroDivisionError:**

print("Error: You cannot divide by zero.")

**except ValueError:**

print("Error: Invalid input, please enter a valid integer.")

**8. Write code for the following scenario and add a try-exception block.**

**a) Program to divide two numbers**

**try:**

a = int(input("enter 1st number"))

b = int(input("enter 2nd number"))

result = a/b

print("the division of a/b is", result)

**except ValueError:**

print("Error: Invalid input, please enter a valid integer.")

**b) Program to convert a string to an integer**

string = input("Enter anything: ")

try:

integer = int(float(string))

print(f"{integer} is an integer.")

except ValueError:

print("Error: The input cannot be converted to an integer.")

**c) Program to access an element in a list**

my\_list = [1, 2, 3, 'trex', 'nakul', 7, 8, 9]

try:

access = int(input("Enter an index to access (0 to 7): "))

access\_element = my\_list[access]

print("Accessed element is:", access\_element)

except IndexError:

print("Error: Accessed element is out of range.")

except ValueError:

print("Error: Please enter a valid integer.")

**d) Program to handle a specific exception**

**try:**

a = int(input("enter 1st number"))

b = int(input("enter 2nd number"))

result = a/b

print("the division of a/b is", result)

**except ValueError:**

print("Error: Invalid input, please enter a valid integer.")

**e) Program to handle any exception**

try:

number = int(input("Enter a number: "))

result = 10 / number

except:

print("An error occurred.")