**Exception Handling 1**

**Python Assignment**

**12 Feb 2023**

**Q1. What is an Exception in python? Write the difference between Exceptions and syntax errors?**

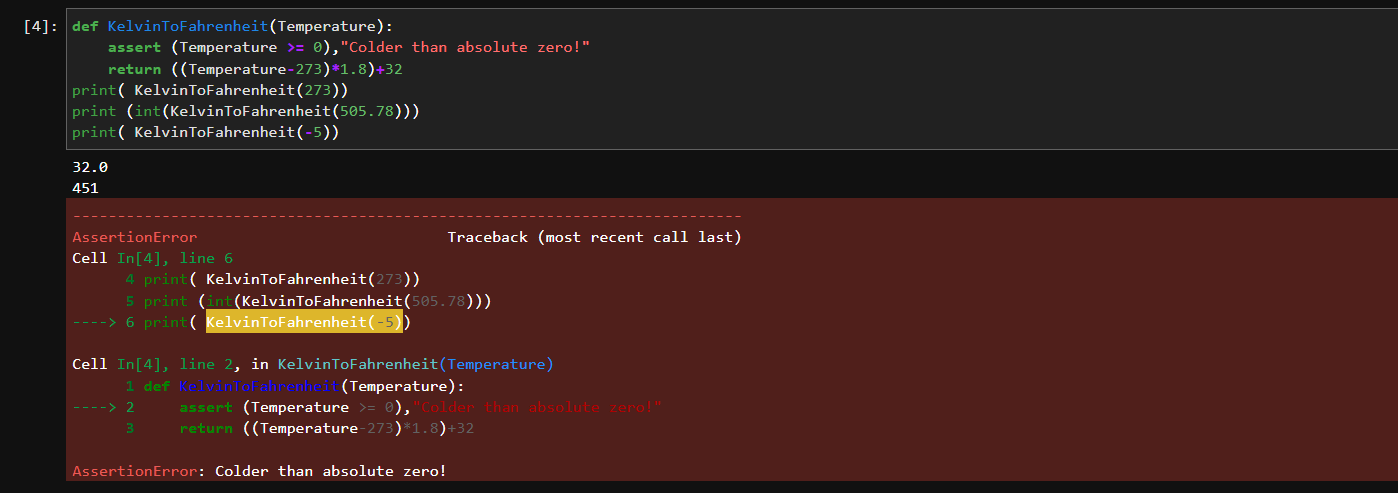
**Sol.** An exception is an event, which occurs during the execution of a program that disrupts the normal flow of the program's instructions. In general, when a Python script encounters a situation that it cannot cope with, it raises an exception. An exception is a Python object that represents an error.

***Syntax errors*** are perhaps the most common kind of complaint you get while you are still learning Python.

Even if a statement or expression is syntactically correct, it may cause an error when an attempt is made to execute it. Errors detected during code execution are called **exceptions**and are not unconditionally fatal. Most exceptions are not handled by programs

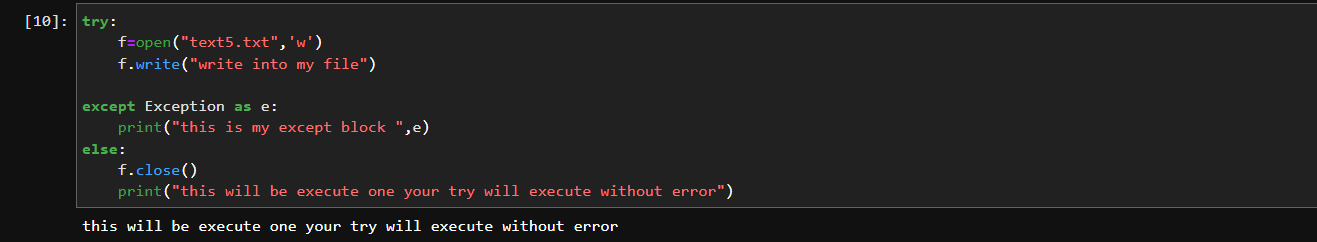
**Q2. What happens when an exception is not handled? Explain with an example.**

**Sol.** If the assertion fails, Python uses ArgumentExpression as the argument for the AssertionError. AssertionError exceptions can be caught and handled like any other exception using the try-except statement, but if not handled, they will terminate the program and produce a traceback.



**Q3. Which Python statements are used to , catch and handle exceptions? Explain with an example.**

**Sol.** The try and except block in Python is used to catch and handle exceptions.

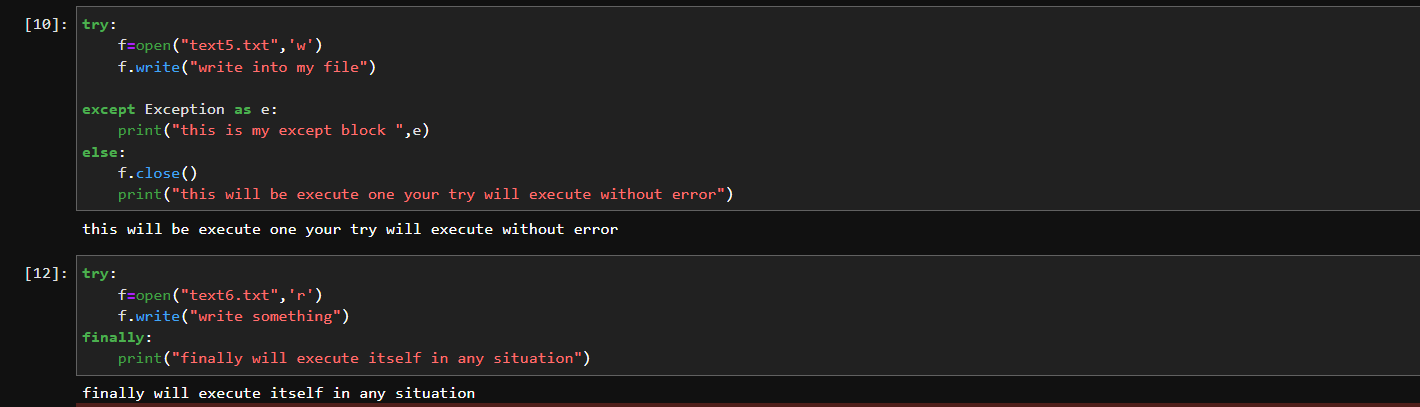


**Q4. Explain with an example:**

* **try and else**
* **Finally**
* **Raise**

**Sol.** **Try** block will test the excepted error to occur. **Else** If there is no exception then this block will be executed

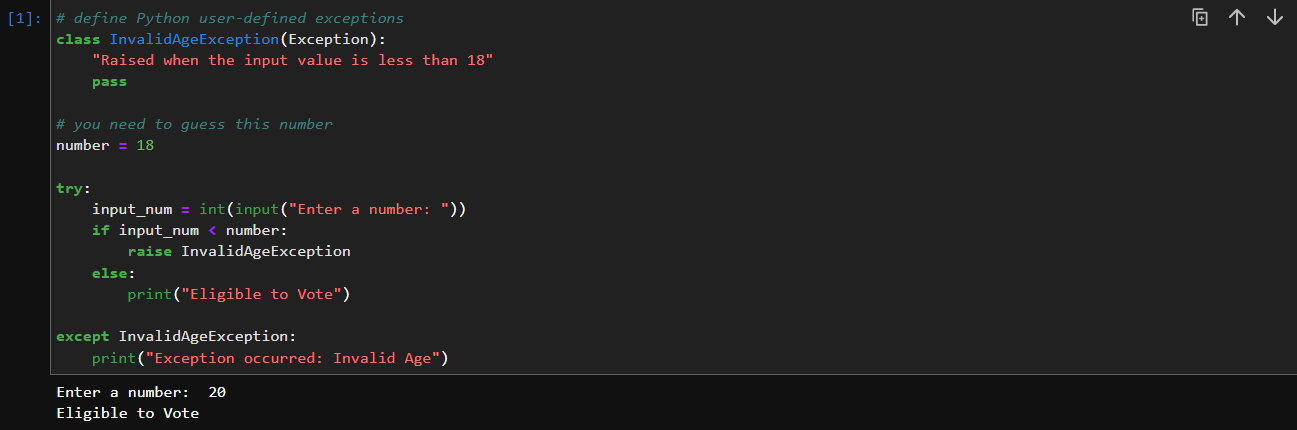
**Finally** block always gets executed either exception is generated or not.



**Q5. What are Custom Exceptions in python? Why do we need Custom Exceptions? Explain with an example.**

# Sol. Exceptions are just regular classes that inherit from the exception class. This makes it super easy to create our own custom exceptions, which can make our programs easier to follow and more readable. An exception need not be complicated, just inherit from exception.

Having custom exceptions - tailored to your specific use cases and that you can raise and catch in specific circumstances - can make your code much more readable and robust, and reduce the amount of code you write later to try and figure out what exactly went wrong.



**Q6. Create custom exception class. Use this class to handle an exception.**

**Sol.**

