CoGrammar

Week 4 – Open Class 1

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Software Engineering Lecture Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
  (FBV: Mutual Respect.)
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Open Classes. You can submit these questions here: **Open Class Questions**

Software Engineering Lecture Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**
- Report a **safeguarding** incident: **www.hyperiondev.com/safeguardreporting**
- We would love your **feedback** on lectures: **Feedback on Lectures**

Progression Criteria

✓ Criterion 1: Initial Requirements

- Complete 15 hours of Guided Learning Hours and the first four tasks within two weeks.

✓ Criterion 2: Mid-Course Progress

- Software Engineering: Finish 14 tasks by week 8.
- Data Science: Finish 13 tasks by week 8.

✓ Criterion 3: Post-Course Progress

- Complete all mandatory tasks by 24th March 2024.
- Record an Invitation to Interview within 4 weeks of course completion, or by 30th March 2024.
- Achieve 112 GLH by 24th March 2024.

✓ Criterion 4: Employability

- Record a Final Job Outcome within 12 weeks of graduation, or by 23rd September 2024.

Lecture Objectives

1. **Review Try-Except Blocks.**
2. **Try-Except examples.**
3. **Open floor: Q&A**

Defensive Programming

- ★ Programmers anticipate errors.
  - ★ User errors
  - ★ Environment errors
  - ★ Logical errors

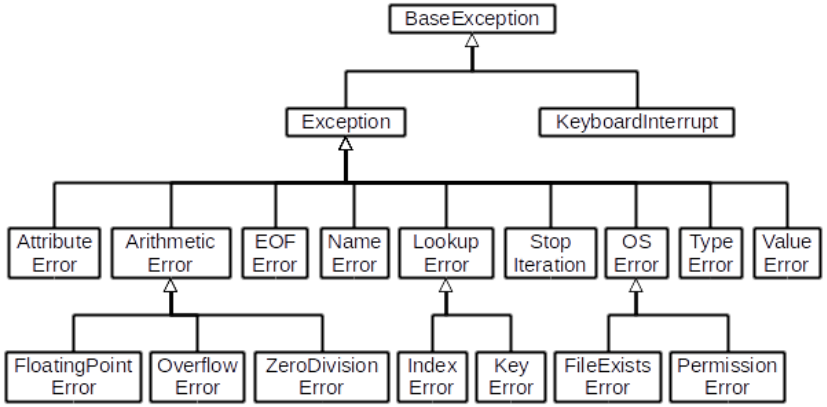
- ★ Code is written to ensure that these errors don’t crash the code base.

- ★ Two ways - if statements and try-except blocks.

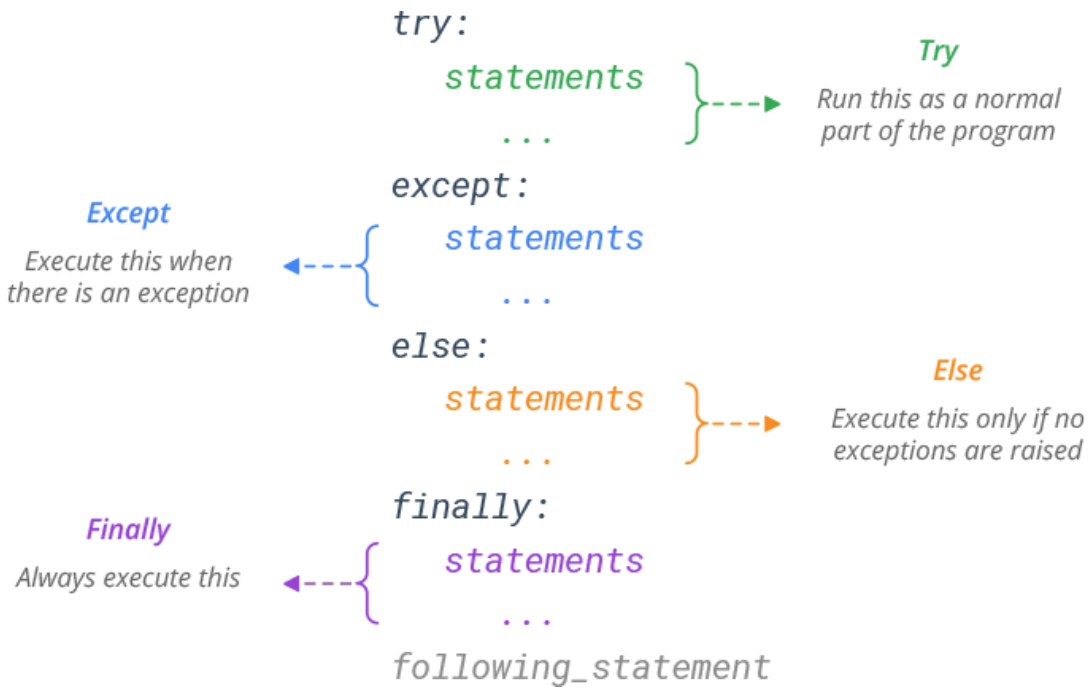
What are exceptions?

An exception is an event, which occurs during the execution of a program that disrupts the normal flow of the initial instructions.

Basic types of exceptions



try-except block structure



https://www.learnbyexample.org/python-exceptions-try-except/

Raising Exceptions

- There will be occasions when you want your program to raise a custom exception whenever a certain condition is met.

- In Python we can do this by using the “raise” keyword and adding a custom message to the exception: In the next example we’re prompting the user to enter a value> 10. If the user enters a number that does not meet that condition, an exception is raised with a custom error message.

Raising Exceptions

```
num = int(input("Please enter a value greater than 10 : "))

if num < 10:

    raise Exception(f"Your value was less than 10. The value of num was : {num}")
```

A Note on try-except

- ★ It may be tempting to wrap all code in a try-except block. However, you want to handle different errors differently.
- ★ Don’t try to use try-except blocks to avoid writing code that properly validates inputs.
- ★ The correct usage for try except should only be for “exceptional” cases. Eg: The potential of Division by 0.

CoGrammar

Refer to python files for revision.

CoGrammar

Questions and Answers

CoGrammar

Thank you for joining