

CSC 148: Introduction to Computer Science

Week 4

Abstraction, abstract data types

Stack implementation and client functions

Handling exceptions



University of Toronto Mississauga,
Department of Mathematical and Computational Sciences



pop()

- Pop the item off the top of the stack
- What happens when the stack is empty?
 - Code should raise an exception
 - But, client code should not crash
 - Handle this gracefully => handling exceptions

SIDE-NOTE:

Raising exceptions

How can we immediately report to client code that one of our functions was called incorrectly?



University of Toronto Mississauga,
Department of Mathematical and Computational Sciences



Recall: strategies for handling bad inputs

- Preconditions (“it’s the user’s fault”)
- Do nothing (“fail silently”)
- Input processing (“fix the problem for them”)



Exceptions

- An **exception** is a special object in Python that represents some kind of error
- **Raising an exception** is a way to **interrupt** the normal execution of a program. The exception object is used to report the type of error, and relevant details



Exceptions

- Generic Exception class
- Other predefined exceptions
 - e.g., `NotImplementedError`, `IndexError`, etc.
- May define our own custom exceptions
 - Can subclass `Exception` to define new custom exceptions
- Let's create our own exception "`EmptyStackError`" and use it when we try to `pop()` from an empty stack



Worksheet

- More operations with stacks
 - stack size ...