

CSC148 fall 2013
inheritance and exception
week 3

Danny Heap
heap@cs.toronto.edu
BA4270 (behind elevators)
<http://www.cdf.toronto.edu/~heap/148/F13/>
416-978-5899

September 24, 2013

Outline

specialize software

raising exceptions

specialize flexibly

If we decided to extend the features of Stack, what's wrong with:

- ▶ modifying the existing Stack?) can't use Stack for non-int
- ▶ cut-paste-modify Stack → MyStack?) changes not propagated
- ▶ include Stack attribute in new classes
different, perhaps feasible



class declaration



we subclass (extend) a superclass (base class) by:

- ▶ declaring that we're extending it...

```
class NewClass(OldClass):
```

```
...
```

- ▶ *modify existing* *add attributes also* **add** methods and attributes to specialize

- ▶ other methods and attributes are searched for in superclass

override versus extend

you may replace or modify old code

- ▶ subclass method with the same name replace superclass method
- ▶ access superclass method with `OldClass.method(self, ...)`
- ▶ `__init__` is a special case — careful

richer communication

return types are not appropriate in all cases

- ▶ what's wrong with `IntStack` returning a “special” integer for `pop-on-empty`?
- ▶ `push` usually has return type `None`, but what if stuff happens?
- ▶ what if the calling code doesn't know what to do?