

# Lecture 31: Exceptions

---

CS 51G  
Spring 2018  
Kim Bruce

# Announcements

- Exercise 15.5.3
- Simon lab Friday
- Courses for next year
  - CS 52 & 55 vs 54
  - For most, CS 54 should be best choice

# Strings are Lists!

- ... but immutable. Can get individual elements via `at(i)`
- Can also iterate through characters in string
  - Does a string represent a number?

```
method isInteger(word:String) -> Boolean {  
  for(word) do {letter: String ->  
    if (("0" > letter) || (letter > "9")) then {  
      return false  
    }  
  }  
  true  
}
```

# Strings are ordered

- `<`, `<=`, `>`, `>=`, `==`, and `!=` work as expected (alphabetical order).
- `ord` gives numeric code of first letter of string

# Using the Exception

```
method changeColor -> Done {
  var newColor: Color
  try {
    newColor := color.r(redField.number)
                    g(greenField.number)
                    b(blueField.number)
  } catch {
    ex: ColorOutOfRange ->
      print "Enter values between 0 and 255 for colors"
      newColor := black
  }
  background.color := newColor
}
```

See <http://www.cs.pomona.edu/classes/cs051G/demos/ColorMixerExceptional/ColorMixer2.grace>

# Using Parameter ex

- Here are some of its methods:
  - `exception` → `exceptionKind`, answers the `exceptionKind` of this exception.
  - `message` → `String`, the message that was provided when this exaction was raised.
  - `data` → `Object`, answers the data object that was associated with this exception when it was raised, if there was one. Otherwise, answers the string “no data”.
  - `lineNumber` → `Number`, the source code line number of the request of raise that created this exception.
  - `backtrace` → `List[[String]]`, a list of strings describing the call stack at the time that this exception was raised. `backtrace.first` is the initial execution environment; `backtrace.last` is the context that raised the exception.

# Another Example

```
def myList:List[[Number]] = list[[Number]] [5,7,9]
var index := 1
try {
  while {index < 7} do {
    print(myList.at(index))
    index := index + 1
  }
} catch {ex: BoundsError ->
  print "went too far!"
  print ("on line {ex.lineNumber} of {ex.moduleName}, {ex.message}")
  print "\n\nBacktrace: {ex.backtrace}"
}
```

Code in

<http://www.cs.pomona.edu/classes/cs051G/demos/ExceptionTest/exceptionTest.grace>

Questions?