Semantics (X)

Exceptions

- The formal semantics of exceptions is out of the scope of this course.
- We will describe the flow of execution when an exception is encountered.

- Java example: A function can throw an exception it does not catch. To do this, it must declare the property of the exception to calling functions by including a throw clause in its preamble.
  - Java permits multiple catch blocks differentiated by their argument.
  - The try/catch structure can also include a block labeled with finally, which has no arguments. Code in the finally section is always executed whether or not the code throws an exception. It is executed even if one of the try/catch blocks calls break or return. [Show trycatch.java, ask the outputs]

```java
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class trycatch {
    public static void main(String[] args) throws IOException {
        int number;
        while (true) {
            try {
                BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
                System.out.print("Enter number: ");
                number = Integer.parseInt(in.readLine());
                if (number == 0) break;
            } catch (NumberFormatException e) {
                System.out.println("Illegal number");
            } finally {
                System.out.println("In the finally block");
            }
        }
        System.out.printf("number is %d\n", number);
    }
}
```

$ javac trycatch.java
$ java trycatch
Enter number: 1
In the finally block
Enter number: 2
In the finally block
Enter number: 3
In the finally block
Enter number: 4
In the finally block
Enter number: 0
In the finally block
number is 0
- Python example: If an exception occurs in the `try` statement, execution moves to the `except` block if the exception matches one of the exceptions listed.
  - The syntax permits multiple `except` cases. An `except` block can also list multiple exceptions inside a tuple, e.g. (RuntimeError, TypeError, NameError, ValueError).
  - An `except` block with no arguments catches all exceptions not explicitly caught by other `except` cases.
  - The equivalent of `throw` in Python is the `raise` keyword. The `raise` statement `takes the exception class as an argument`. [Show trycatch.py]
  - Python also includes a `finally` clause like Java.

```python
def demo1():
a = 0

    while a == 0:
        try:
            s = input('enter a number: ')
            val = int(s)

        except ValueError:
            print("not a valid number")
            continue

        if val == 0:
            a = 1

        print("terminating")

def demo2_helper():
    raise 1

def demo2():
    try:
        demo2_helper()
    except:
        print("demo 2 catching the error")

demo1()
#demo2()
```

Comment out `demo2()` and run the code

```bash
$ python3 trycatch.py
enter a number: a
not a valid number
enter a number: 1
enter a number: 0
terminating
```

Comment out `demo1()`, and run again

```bash
$ python3 trycatch.py
demo 2 catching the error
```