# How to handle Exceptions

* When an error occurs during runtime, it can be referred to as *throwing an exception*.
* When an exception is thrown, Python ends the program and prints information about the exception to the console. The last line of this information includes the type of exception and a brief description of the exception.
* Some exceptions occur due to programming errors. You need to fix these errors before the program is ready to use.
* Some exceptions occur due to causes outside of the program. These exceptions need to be *handled* by your Python code so the program doesn’t crash when they occur.

1. There is a list of exceptions posted in your d2l shell. Refer to this list and tell which kind of exception(s) could be caused by the following statements:

|  |  |
| --- | --- |
| Statement | Type of exception |
| 1. number = int(input(“Enter an integer: “)) | ValueError |
| 2. price = float(input(‘Enter price: “)) | ValueError |
| 3. Ans = 1/0 | ZeroDivisionError |
|  |  |
| 4. a = 13 # a has not been defined |  |
| 5. import reading #the reading module does not exist | ModuleNotFoundError |
| 6. for i in range(3):  print (i) |  |
|  |  |
| 7. with open(milesPerGallon.txt, “r”) as file: | 1. |
|  | 2. OSError (file found – error reading file) |

2. Using statements 1 to 3 above, write a try: / except: statement that catches an exception. Watch your indentation.

|  |
| --- |
|  |
|  |
|  |

3. Statement 7, above, involves opening a file to be read. This can cause 2 specific errors. Also, because a file is being opened, it ultimately needs to be closed. Create a try: / except:/ finally: statement to deal with this type of exception.

|  |
| --- |
|  |