

# 10 Team Activity: Handling Exceptions

## Instructions

---

Work as a team as explained in the instructions for the [lesson 2 team activity](#).

## Problem Statement

---

Every year in the United States, thousands of people are killed and millions more are injured in traffic accidents on the nation's roads and highways. The U.S. National Highway Traffic Safety Administration (NHTSA) maintains [data about the accidents](#), including a list of factors contributing to each accident.

The CSV file `accidents.csv` contains summary data from the NHTSA about the accidents that occurred on U.S. roads and highways during the years 2010 through 2017. `accidents.py` is a Python program that uses the data to estimate how many lives would have been saved and injuries avoided if drivers had stopped dangerous behavior such as speeding and using cell phones while driving.

Unfortunately, `accidents.py` does not contain any code to validate the input that the user enters while the program is running. Add exception handling code to `accidents.py` to validate the user's input and print helpful error messages when the user enters incorrect input.

## Helpful Documentation

---

The [prepare content](#) for this lesson explains how to wrap `try` and `except` around a normal block of code. It also includes an example that shows how to [validate user input](#) within a loop.

This [video about error handling](#) (17 minutes) shows a BYU-Idaho faculty member adding a `try` block and `except` blocks to a program to handle errors that might occur while the program is executing.

[open function](#), [FileNotFoundError](#), and [PermissionError](#)

[float function](#) and [ValueError](#)

[ZeroDivisionError](#) and [csv.Error](#)

# Assignment

---

Do the following:

1. Download and save both [accidents.csv](#) and [accidents.py](#) in the same folder.
2. Open `accidents.csv` in VS Code or another editor and notice that it contains ten columns and eight rows of data regarding traffic accidents in the United States.
3. Open `accidents.py` in VS Code and notice that lines 21–22 contain code to ask the user for a filename and then open that file. It is possible that when the computer executes line 21, the user may enter the name of a file that does not exist or may enter the name of a file that the user does not have permission to read. These two situations cause the computer to raise a `FileNotFoundError` OR `PermissionError` when the computer tries to open the file at line 22.
4. Notice that lines 25–26 of `accidents.py` contains code to prompt the user for a number. It is possible that while the computer is executing lines 25–26, the user may enter text that is not a number, for example: "4ro". This situation causes the computer to raise a `ValueError`. It is also possible that the user may enter a number that is too low or too high, for example: -15 or 701234932. In this situation, the computer does not raise an exception, but this situation is still a mistake that the program should alert the user about.
5. Notice that lines 45–55 of `accidents.py` contain code to process each line of the CSV file. It is possible that the CSV file is formatted incorrectly which would cause the computer to raise `csv.Error`. It is also possible that one of the rows in the CSV file contains zero (0) in the "Fatal Crashes" column. This would cause the computer to raise `ZeroDivisionError` at line 75 within the `estimate_reduction` function.

## Core Requirements

1. Add exception handling code that prints a useful error message when the computer raises `FileNotFoundError` OR `PermissionError`.
2. Add exception handling code that prints a useful error message when the computer raises `ValueError` while it is converting the percentage entered by the user to a float.
3. Add code that prints a useful error message when the user enters a percentage that is less than 0 or greater than 100. Hint: you could add an `if` statement immediately after the line of code that gets the percentage from the user.

## Stretch Challenges

If your team finishes the core requirements in less than an hour, complete one or more of these stretch challenges. Note that the stretch challenges are optional.

1. Add exception handling code to handle `ZeroDivisionError` that might be raised by the `estimate_reduction` function. This exception handling code should print a useful error message that includes the filename and line number of the line that contains zero (0) in the "Fatal Crashes" column.
2. Add exception handling code to handle `csv.Error` that might be raised by the `csv` module. This exception handling code should print a useful error message that includes the filename and line number of the line that is formatted incorrectly.
3. Test *all* of your error handling code, including the code that handles `ZeroDivisionError` and `PermissionError`. To test `ZeroDivisionError`, you could temporarily change one of the numbers in the "Fatal Crashes" column of `accidents.csv` to zero (0). To test `PermissionError`, you could create a CSV file or a simple text file and then use File Explorer or Finder to change the permissions on that file so that your account doesn't have permission to read that file.

## Testing Procedure

---

Verify that your program works correctly by following each step in this testing procedure:

1. Run the `accidents.py` program and enter the inputs shown below. Verify that the exception handling code that you added to `accidents.py` prints error messages as shown below.

```
> python accidents.py
Name of file that contains NHTSA data: dentiaccs.vsc
[Errno 2] No such file or directory: 'dentiaccs.vsc'
Please choose a different file.

> python accidents.py
Name of file that contains NHTSA data: unreadable.csv
[Errno 13] Permission denied: 'unreadable.csv'
Please choose a different file.

> python accidents.py
Name of file that contains NHTSA data: accidents.csv
Percent reduction of texting while driving [0, 100]: 4r0
Error: could not convert string to float: '4r0'

> python accidents.py
Name of file that contains NHTSA data: accidents.csv
Percent reduction of texting while driving [0, 100]: -5
```

```
Error: -5.0 is too low. Please enter a different number.
```

```
> python accidents.py
```

```
Name of file that contains NHTSA data: accidents.csv
```

```
Percent reduction of texting while driving [0, 100]: 110
```

```
Error: 110.0 is too high. Please enter a different number.
```

```
> python accidents.py
```

```
Name of file that contains NHTSA data: accidents.csv
```

```
Percent reduction of texting while driving [0, 100]: 50
```

With a 50.0% reduction in using a cell phone while driving, approximately the following number of injuries and deaths would have been prevented in the USA.

Year	Injuries	Deaths
2010	15631	230
2011	15811	232
2012	16759	240
2013	15738	224
2014	15052	211
2015	17006	247
2016	19953	246
2017	16077	217

## Sample Solution

---

Please work diligently with your team for the one hour meeting. After the meeting is over, please compare your solution to this [sample solution](#) [↓]. Please **do not look at the sample solutions** until you have either finished the program or diligently worked for at least one hour. At the end of the hour, if you are still struggling to complete the assignment, you may use the sample solution to help you finish.

## Submission

---

When you have finished the activity, please report your progress via the associated I-Learn quiz. When asked about which of the requirements you completed, feel free to include any work done during the team meeting or after the meeting, including work done with the help of the sample solution, if necessary. In short, report on what you were able to accomplish, regardless of when you completed it or if you needed help from the sample solution.