



Lesson 1.5: Exceptions

<https://codehs.com/course/21079/lesson/1.5>

Description	In this lesson, students explore Python's way of handling errors with exceptions.
Objective	Students will be able to: <ul style="list-style-type: none">• create programs that can gracefully handle exceptions• continue to function when an error is raised
Activities	1.5.1 Video: Exceptions 1.5.2 Check for Understanding: Exceptions 1.5.3 Example: Enter a Number 1.5.4 Example: Enter Name and Age 1.5.5 Exercise: Temperature Converter, Part 2 1.5.6 Exercise: Enter a Positive Number
Prior Knowledge	<ul style="list-style-type: none">• Functions• Parameters• Return Values
Planning Notes	<ul style="list-style-type: none">• This lesson contains an accompanying worksheet. Determine if and how this worksheet will be used and print the necessary number of copies.
Standards Addressed	
Teaching and Learning Strategies	<p>Lesson Opener:</p> <ul style="list-style-type: none">• Have students brainstorm and write down answers to the discussion questions listed below. Students can work individually or in groups/pairs. Have them share their responses. [5 mins] <p>Activities:</p> <ul style="list-style-type: none">• Watch video as a class or individually and have students complete the quiz. [7-9 mins]• Students explore the <i>Enter a Number</i> example. [2-3 mins]<ul style="list-style-type: none">◦ <i>Optional Extension:</i> Try entering the following values. What do you expect to happen and what is the result?<ul style="list-style-type: none">▪ 2 (Program displays number)▪ -4 (Program displays number)▪ .5 (Program displays message telling user they did not enter an integer)▪ hi (Program displays message telling user they did not enter an integer)• Students explore the <i>Enter Name and Age</i> example. [2-3 mins]<ul style="list-style-type: none">◦ <i>Optional Challenge:</i> Alter the program so that the user is continually asked for their age until they enter a valid value.• Students complete the <i>Divisibility, Part 2</i> exercise individually. [3-5 mins]• Students complete the <i>Temperature Converter, Part 2</i> exercise individually. [5-7 mins]• Students complete the <i>Enter a Positive Number</i> exercise individually. [5-7 mins]

	<ul style="list-style-type: none"> Students complete the <i>Errors in Python</i> handout individually or in pairs or for homework, as time permits. [7-15 mins] <p>Lesson Closer:</p> <ul style="list-style-type: none"> Have students reflect and discuss their responses to the end of class discussion questions. [5 mins]
<p>Discussion Questions</p>	<p>Beginning of Class: Have students work in their Sandbox to develop some basic programs, such as:</p> <ul style="list-style-type: none"> Ask the user for an integer and add 1 to that number. Print the total. <ul style="list-style-type: none"> Instruct students to enter 1, .5 -1, and 'hi' as user input. What happens when they try these inputs? <ul style="list-style-type: none"> <i>They should notice that .5 and 'hi' cannot be converted to an 'int'. Have them note the type of error they receive (ValueError here).</i> Ask the user for an integer and divide 10 by that value. <ul style="list-style-type: none"> Instruct students to enter 1, .5 -1, 0, and 'hi' as user input. What happens when they try these inputs? <ul style="list-style-type: none"> <i>They should notice that .5 and 'hi' cannot be converted to an 'int', throwing 'ValueError's. They should also note that an input of 0 throws a 'ZeroDivisionError'.</i> <p><i>When these programs fail, discuss the implications with students: What if our program was actually controlling a space shuttle, and a simple math mistake (like accidentally dividing by zero) caused our whole program to crash?</i></p> <p>End of Class:</p> <ul style="list-style-type: none"> What types of errors have we seen so far? When do they occur? <ul style="list-style-type: none"> <i>ZeroDivisionError: occurs when attempting to divide a value by zero, ValueError: occurs when the program expects one value type and another is given.</i> Why is it bad to let our entire program crash? <ul style="list-style-type: none"> <i>If our program crashes, the user will never be able to get to the expected output. We'd much rather have a way around the error so the user can be given a second (or third) chance to enter the needed information in the correct format.</i> What are some examples of error handling in real life? (A great example here is what happens when your GPS tells you to turn down a street that is closed for construction: Does the car stop working, or can the driver catch this error and work around it?) <ul style="list-style-type: none"> <i>Expecting to take the elevator and encountering a sign that says 'Elevator out of service' so instead, you take the stairs, etc. (Answers may vary.)</i>
<p>Resources/Handouts</p>	<p>Errors in Python (teacher)</p> <p>Errors in Python (student)</p> <p>Exceptions Examples Exploration (teacher)</p> <p>Exceptions Examples Exploration (student)</p>

Vocabulary

Term	Definition	
Modification: Advanced	Modification: Special Education	Modification: English Language Learners

- Pair programming with another student
- Print out video slides for students to reference

- Print out video slides and have students use dictionary to translate unknown vocabulary
- Pair programming with another student