



Name _____ Class _____ Date _____

AFTER THIS WORKSHEET YOU CAN

✓ Catch errors with try/except ✓ Use else and finally ✓ Raise exceptions ✓ Name common error types



Section A • Concept Check

• BEGINNER

4 × 1 = 4

- 1 Errors are caught with:
☐ A. catch ☐ B. except ☐ C. rescue ☐ D. trap
- 2 `int('abc')` raises:
☐ A. `TypeError` ☐ B. `ValueError` ☐ C. `KeyError`
☐ D. `OSError`
- 3 `finally` runs:
☐ A. only on error ☐ B. only on success
☐ C. always ☐ D. never
- 4 Raise your own error with:
☐ A. `throw` ☐ B. `raise` ☐ C. `error()` ☐ D. `panic`



Section B • Problem Solving

• INTERMEDIATE

2 + 3×3 = 11

- 5 Risky code goes in the _____ block.
- 6 `1/0` raises _____.
- 7 Wrap `int(input())` in `try/except ValueError`.

- 8 What error does `[1,2][5]` raise?

- 9 When does `else` (after `try`) run?



Section C • Challenge

• CHALLENGE

5

- 10 Why catch exceptions?

🧠 Reflection — the most useful thing I learned: _____

A ___/4 B ___/11 C ___/5 Total ___/20

Teacher's Signature _____

Parent's Signature _____

✂️ ANSWER KEY — FOLD OR CUT BEFORE HANDING OUT

1-B 2-B 3-C 4-B | 5 `try` 6 `ZeroDivisionError` 7 = `try: n=int(input()) except ValueError: print('Not a number')` 8 = `IndexError` 9 = Only if no exception occurred | 10 = To handle failures gracefully instead of crashing