Debugging Report

- What Was Wrong?

I found a main problem in the code - it breaks when trying to process the Tablet price. This happens because the Tablet's price is written as "500" (with quotes) which makes it a string, while all the other prices are numbers. When the code tries to multiply "500" by 0.2 to calculate the discount, Python gets confused and crashes.

- How I Found It:

1. I ran the code and saw it crash

2. Used VSC debugging tool to see where the code was but that didn’t help much since it kept crashing. so looked over the code and noticed the quote which threw me off. I changed it back into a number by removing the quotes, and the code worked.

- How I Fixed It:

1. Added a way to check if numbers are valid

2. Made sure the code doesn't crash if something goes wrong

3. Added helpful error messages so we know what went wrong

Here's the fixed code:

```python

# FIXED CODE Debugging Exercise

# This code is the fixed and improved version of the broken code

def check\_number(number, name):

# Makes sure a number is valid

try:

num = float(number)

if num < 0:

raise ValueError(f"{name} can't be negative!")

return num

except:

raise ValueError(f"Hey, {name} needs to be a real number!")

def calculate\_discount(price, discount\_rate):

# Figures out how much discount to give

price = check\_number(price, "price")

discount\_rate = check\_number(discount\_rate, "discount rate")

if discount\_rate > 1:

raise ValueError("Discount rate should be less than 1 (like 0.2 for 20%)")

return round(price \* discount\_rate, 2)

def apply\_discount(price, discount\_amount):

# Takes off the discount from the price

price = check\_number(price, "price")

discount\_amount = check\_number(discount\_amount, "discount amount")

return round(price - discount\_amount, 2)

def main():

# List of products and their discounts

products = [

{"name": "Laptop", "price": 1000, "discount\_rate": 0.1},

{"name": "Smartphone", "price": 800, "discount\_rate": 0.15},

{"name": "Tablet", "price": 500, "discount\_rate": 0.2},

{"name": "Headphones", "price": 200, "discount\_rate": 0.05}

]

# Go through each product and calculate its discount

for product in products:

try:

price = product["price"]

discount\_rate = product["discount\_rate"]

print(f"\nCalculating discount for: {product['name']}")

discount = calculate\_discount(price, discount\_rate)

final\_price = apply\_discount(price, discount)

print(f"Original Price: ${price}")

print(f"You save: ${discount}")

print(f"Final Price: ${final\_price}")

except ValueError as e:

print(f"\nOops! Problem with {product['name']}: {str(e)}")

except:

print(f"\nSomething went wrong with {product['name']}!")

if \_\_name\_\_ == "\_\_main\_\_":

main()

```

- How I Tested It:

1. Ran the code with all the products

2. Made sure it could handle the string "500" for the Tablet (But I took them away even though it can handle it)

3. Checked that all the discounts were calculated correctly

4. Tried some negative numbers to make sure it caught them

5. Added some invalid inputs to make sure it didn't crash

What's Better Now:

- The code doesn't crash anymore

- It gives friendly error messages

- It rounds prices to 2 decimal places (since we are working with money)

- It keeps going even if one product has a problem

- It catches negative numbers and tells you they're not allowed