NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES

FAST - PESHAWAR CAMPUS

Subject: Software Construction and Development Lab (CL-3001)

Instructor: Muhammad Saood Sarwar

Assignment 2 **Section: B**

Name: _____ **Roll No: _____**

Below is the exception handling code. Display the output for each scenario, write ERROR if any.

Question 1

```
try:
    print("Starting the checkout process...")
    cart_total = input("Enter the total cost of your shopping cart: ")

    #hint: string.replace(old, new, count)
    if not cart_total.replace('.', '', 1).isdigit():
        raise ValueError("Invalid input string: Please enter a numeric value")

    cart_total = float(cart_total)

    if cart_total <= 0:
        raise ValueError("Total cost must be greater than zero")

    discount = 0.1
    final_amount = cart_total - (cart_total * discount)
    print(f"Final amount after 10% discount: {final_amount}")

except ValueError as ve:
    print(f"Caught ValueError: {ve}")

except Exception as e:
    print(f"Caught an unexpected error: {e}")

finally:
    print("Checkout process completed.")
```

Input: 100.0	**Input: 100.0.0**

Question 2

```
try:
    num = 1.0
    num = int(num)
    result = "Hello" + num
except (ValueError, TypeError) as e:
    print(f"Caught an error: {type(e).__name__}")
```

Question 3

```
data = [10, 20, 30]
try:
    value = data[10]
except:
    pass
except IndexError:
    print("Index out of range!")
finally:
    print("Program completed successfully.")
```

Question 4

```
class CustomError(Exception):
    pass
try:
    raise CustomError("This is a custom error")
except CustomError as ce:
    print("Caught:", ce)
finally:
    print("Final block executed")
```

Question 5

```
class InvalidOperationError(Exception):
    pass
def divide(a, b):
    assert b != 0, "Cannot divide by zero"
    if b == 1:
        raise InvalidOperationError("Cannot divide by one.")
    return a / b
try:
    numerator = float(input("Enter numerator: "))
    denominator = float(input("Enter denominator: "))
    result = divide(numerator, denominator)
    print(f"Result: {result}")
except (AssertionError, InvalidOperationError) as e:
    print(e)
except ValueError:
    print("Invalid input.")
finally:
    print("Execution completed.")
```

```
numerator: a              denominator: 0
```