Exception Handling in Python – Practice Questions (CBSE Class XII)

Section A: Very Short Answer (1 Mark)

1. What is an exception in Python?

2. Name any two built-in exceptions in Python.

3. Which block always executes in exception handling?

4. What is the use of the raise keyword?

Section B: Output Based Questions

```
1.
try:
    a = 5
    b = 0
    print(a/b)
except ZeroDivisionError:
    print("Error")
else:
    print("Done")
```

```
2.
try:
    x = int("Hello")
    print(x)
except ValueError:
    print("Invalid Value")
```

Section C: Find the Error

```
1.
try:
    print(10/2)
finally:
    print("End")
```

```
2.
try:
    print(a)
except NameError
    print("Error")
```

Section D: Case Study Based Question

A school software accepts student marks. Sometimes users enter text instead of numbers or zero as total marks. This causes program failure.

```
try:
    marks = int(input("Enter marks: "))
    total = int(input("Enter total marks: "))
    percentage = (marks/total)*100
    print(percentage)
except:
    print("Invalid input")
```

a) Name two exceptions that may occur in the above code.
b) Rewrite the code using specific exception handling.

Section A: Very Short Answer (1 Mark)

1. **What is an exception in Python?** An exception is a runtime error that occurs during the execution of a program, which disrupts the normal flow of the program's instructions.
2. **Name any two built-in exceptions in Python.** Common examples include `ZeroDivisionError` and `ValueError`.
3. **Which block always executes in exception handling?** The `finally` block always executes, whether an exception occurred or not.
4. **What is the use of the `raise` keyword?** It is used to manually trigger or "throw" an exception when a specific condition is met in your code.

Section B: Output Based Questions

1. **Output:** `Error` *Explanation: Dividing 5 by 0 triggers a `ZeroDivisionError`. The `else` block is skipped because an error occurred.*
2. **Output:** `Invalid Value` *Explanation: The string "Hello" cannot be converted to an integer, triggering a `ValueError`.*

Section C: Find the Error

1. **Error:** Technically, this code is **syntactically correct**. In Python, a `try` block can be followed by a `finally` block without an `except` block. However, if the goal was to *catch* an error, an `except` block is missing.
2. **Error:** There is a missing **colon (:)** after the exception name. *Corrected line:* *`except NameError:`*

Section D: Case Study Based Question

a) Name two exceptions that may occur in the above code.

- `ValueError` (if the user types letters instead of numbers).
- `ZeroDivisionError` (if the user enters 0 for total marks).

b) Rewrite the code using specific exception handling:

```
try:
    marks = int(input("Enter marks: "))
    total = int(input("Enter total marks: "))
    percentage = (marks / total) * 100
    print(percentage)
except ValueError:
    print("Invalid input: Please enter numbers only.")
except ZeroDivisionError:
    print("Invalid input: Total marks cannot be zero.")
```