

SENIOR SECONDARY EXAMINATION, 2024
COMPUTER SCIENCE

Time – 03 Hours 15 Minutes

Marks: 56

GENERAL INSTRUCTIONS TO THE EXAMINEES :

- 1) Candidate must write first his / her Roll No. on the question paper compulsorily.
- 2) All the questions are compulsory.
- 3) Write the answer to each question in the given answer-book only.
- 4) For questions having more than one part, the answers to those parts are to be written together in continuity.
- 5) If there is any error / difference / contradiction in Hindi & English version of the question paper, the question of Hindi version should be treated valid.
- 6) Write down the serial number of the question before attempting it.

Section A (MCQs - 9 Questions, 1 Mark Each)

1. What type of error occurs when Python syntax rules are not followed?

- (a) Syntax Error (b) Runtime Error
(c) Logical Error (d) OverflowError

जब पायथन सिंटैक्स के नियमों का पालन नहीं किया जाता, तो किस प्रकार की त्रुटि होती है?

- (a) सिंटैक्स एरर (b) रनटाइम एरर
(c) लॉजिकल एरर (d) ओवरफ्लो एरर

2. Which built-in exception is raised when attempting division by zero?

किस बिल्ट-इन अपवाद को शून्य द्वारा विभाजन करने का प्रयास करने पर उठाया जाता है?

- (a) ZeroDivisionError (b) ValueError
(c) NameError (d) IndexError

3. The assert statement in Python is used for:

- (a) Validating conditions (b) Handling exceptions
(c) Optimizing code (d) Raising OverflowError

पायथन में assert कथन का उपयोग किस लिए किया जाता है?

- (a) स्थितियों को मान्य करने के लिए (b) अपवादों को संभालने के लिए
(c) कोड को अनुकूलित करने के लिए (d) ओवरफ्लो एरर को उठाने के लिए

4. Which exception is raised when the input() function reaches the end of file?

- (a) EOFError (b) FileNotFoundError (c) IOError (d) TypeError

जब input() फ़ंक्शन फ़ाइल के अंत तक पहुँच जाता है, तो कौन सा अपवाद उठाया जाता है?

- (a) EOFError (b) FileNotFoundError (c) IOError (d) TypeError

5. What is the role of the finally clause in exception handling?

- (a) Executes only if no exception is raised (b) Executes regardless of exceptions
(c) Handles all raised exceptions (d) Logs errors to a file

अपवाद संभालने में finally खंड की भूमिका क्या है?

- (a) केवल तभी निष्पादित होता है जब कोई अपवाद नहीं उठाया जाता है (b) अपवादों की परवाह किए बिना निष्पादित होता है
(c) सभी उठाए गए अपवादों को संभालता है (d) त्रुटियों को फ़ाइल में लॉग करता है

6. What exception is raised if the wrong module is imported?

- (a) ImportError (b) IOError (c) IndexError (d) KeyError

यदि गलत मॉड्यूल आयात किया जाता है, तो कौन सा अपवाद उठाया जाता है?

- (a) ImportError (b) IOError (c) IndexError (d) KeyError

7. What type of error occurs when accessing an undefined variable?

- (a) NameError (b) ValueError (c) IndentationError (d) AttributeError

अपरिभाषित वेरिएबल को एक्सेस करने पर किस प्रकार की त्रुटि होती है?

- (a) NameError (b) ValueError (c) IndentationError (d) AttributeError

8. Which statement is used to raise exceptions explicitly in Python?

- (a) raise (b) assert (c) except (d) pass

पायथन में स्पष्ट रूप से अपवाद उठाने के लिए किस कथन का उपयोग किया जाता है?

- (a) raise (b) assert (c) except (d) pass

9. What type of exception is caused by improper indentation in Python code?

- (a) IndentationError (b) SyntaxError (c) NameError (d) KeyError

पायथन कोड में अनुचित इंडेंटेशन के कारण किस प्रकार का अपवाद होता है?

- (a) IndentationError (b) SyntaxError (c) NameError (d) KeyError
-

(Fill in the Blanks - 4 Questions, 1 Mark Each)

1. _____ is raised when a non-integer is passed to a function expecting an integer.

_____ तब उठाया जाता है जब किसी पूर्णांक की अपेक्षा वाली फ़ंक्शन में गैर-पूरकांक पारित किया जाता है।

2. To handle multiple exceptions, we use multiple _____ blocks.

कई अपवादों को संभालने के लिए, हम कई _____ ब्लॉक का उपयोग करते हैं।

3. The try block is used to write code that _____.

try ब्लॉक का उपयोग उस कोड को लिखने के लिए किया जाता है जो _____।

4. If no exception is raised, the _____ block is executed after the try block.

यदि कोई अपवाद नहीं उठाया जाता है, तो try ब्लॉक के बाद _____ ब्लॉक निष्पादित होता है।

(Very Short Answer Type Questions - 8 Questions, 1 Mark Each)

1. Define runtime errors in Python.

पायथन में रनटाइम त्रुटियों को परिभाषित करें।

2. What is the use of the except block?

except ब्लॉक का उपयोग क्या है?

3. Name two examples of built-in exceptions in Python.

पायथन में दो बिल्ट-इन अपवादों के उदाहरण दें।

4. Differentiate between raise and assert.

raise और assert में अंतर करें।

5. What does the finally block ensure in Python code?

पायथन कोड में finally ब्लॉक क्या सुनिश्चित करता है?

6. Explain the term "exception propagation".

"अपवाद प्रसार" शब्द को समझाएं।

7. Define Syntax Error.

सिंटैक्स एरर को परिभाषित करें।

8. What is the difference between is and == in Python?

पायथन में is और == में क्या अंतर है?

Section B (Short Answer Type Questions - 12 Questions, 1.5 Marks Each)

1. Write a Python program to handle a ZeroDivisionError exception.

ZeroDivisionError अपवाद को संभालने के लिए एक पायथन प्रोग्राम लिखें।

2. What is the difference between a try block and an except block in Python?

पायथन में try ब्लॉक और except ब्लॉक के बीच क्या अंतर है?

3. Explain the purpose of the finally block in exception handling.

अपवाद संभालने में finally ब्लॉक का उद्देश्य समझाएं।

4. Give two examples of syntax errors in Python and explain how they occur.

पायथन में सिंटैक्स एरर के दो उदाहरण दें और उनके होने का कारण समझाएं।

5. Explain the raise statement in Python with an example.

एक उदाहरण के साथ पायथन में raise कथन की व्याख्या करें।

6. Differentiate between NameError and TypeError.

NameError और TypeError के बीच अंतर करें।

7. Write a short note on EOFError with an example.

EOFError पर एक संक्षिप्त टिप्पणी लिखें और उदाहरण दें।

8. What is an IndexError, and when does it occur in Python?

IndexError क्या है, और पायथन में यह कब होता है?

9. How does Python handle multiple exceptions in a single program? Write an example code.

पायथन एक प्रोग्राम में कई अपवादों को कैसे संभालता है? एक उदाहरण कोड लिखें।

10. Write the steps involved in the process of exception handling in Python.

पायथन में अपवाद संभालने की प्रक्रिया में शामिल चरणों को लिखें।

11. Why do we need custom exceptions in Python? Write an example program to create and raise a user-defined exception.

हमें पायथन में कस्टम अपवादों की आवश्यकता क्यों है? एक उपयोगकर्ता-परिभाषित अपवाद बनाने और उठाने के लिए एक उदाहरण प्रोग्राम लिखें।

12. Explain the term "EOF (End of File)".

Section C (Long Answer Type Questions - 3 Questions, 3 Marks Each)

1. Write a Python program to demonstrate the use of try, except, and finally blocks to handle exceptions.

अपवादों को संभालने के लिए try, except, और finally ब्लॉक के उपयोग का प्रदर्शन करने के लिए एक पायथन प्रोग्राम लिखें।

2. Explain the types of Python exceptions (built-in and user-defined) with examples for each.

पायथन अपवादों के प्रकार (बिल्ट-इन और उपयोगकर्ता-परिभाषित) को उदाहरणों के साथ समझाएं।

3. Design a Python program that reads a file and handles the following exceptions:

एक पायथन प्रोग्राम डिज़ाइन करें जो एक फाइल पढ़े और निम्नलिखित अपवादों को संभाले:

- i. FileNotFoundError
- ii. ValueError

Section D (Essay Answer Type Questions - 2 Questions, 4 Marks Each)

1. Write an essay on Python Exception Handling, covering the following points:

- I. Definition of exceptions
- II. Why exception handling is important
- III. The role of try, except, else, and finally clauses
- IV. Example program

OR

Write a Python program to handle exceptions related to user input. Include the following features:

- I. Validate that the input is a number
 - II. Handle division by zero
 - III. Handle non-numeric input
2. Explain the process of raising and catching user-defined exceptions in Python with examples. Include the syntax and benefits of using custom exceptions.

OR

Design a Python program to demonstrate the use of the assert statement. Write an explanation of how assertions can help in debugging.



Answer Key:

MCQ's Answers

1. (a) Syntax Error
2. (a) ZeroDivisionError
3. (a) Validating conditions
4. (a) EOFError
5. (b) Executes regardless of exceptions
6. (a) ImportError
7. (a) NameError
8. (a) raise
9. (a) IndentationError

Fill in the Blanks -

1. **TypeError**

TypeError तब उठाया जाता है जब किसी पूर्णांक की अपेक्षा वाली फ़ंक्शन में गैर-पूरकांक पारित किया जाता है।

2. **except**

कई अपवादों को संभालने के लिए, हम कई **except** ब्लॉक का उपयोग करते हैं।

3. **might raise an exception**

try ब्लॉक का उपयोग उस कोड को लिखने के लिए किया जाता है जो **अपवाद उत्पन्न कर सकता है**।

4. **else**

यदि कोई अपवाद नहीं उठाया जाता है, तो try ब्लॉक के बाद **else** ब्लॉक निष्पादित होता है।

Very Short Answer Type Questions

1. **Define runtime errors in Python.**

Runtime errors are errors that occur during the execution of a program, causing it to terminate abnormally.

पायथन में रनटाइम त्रुटियों को परिभाषित करें।

रनटाइम त्रुटियां वे त्रुटियां हैं जो प्रोग्राम के निष्पादन के दौरान उत्पन्न होती हैं और प्रोग्राम को असामान्य रूप से समाप्त कर देती हैं।

2. **What is the use of the except block?**

The except block is used to handle exceptions and prevent the program from crashing.

except ब्लॉक का उपयोग क्या है?

except ब्लॉक का उपयोग अपवादों को संभालने और प्रोग्राम को क्रैश होने से बचाने के लिए किया जाता है।

3. **Name two examples of built-in exceptions in Python.**

Examples of built-in exceptions: ValueError, TypeError.

पायथन में दो बिल्ट-इन अपवादों के उदाहरण दें।

बिल्ट-इन अपवादों के उदाहरण: ValueError, TypeError।

4. **Differentiate between raise and assert.**

- raise is used to manually trigger an exception.
- assert is used for debugging purposes to check if a condition is true.

raise और assert में अंतर करें।

- raise का उपयोग मैन्युअली अपवाद उत्पन्न करने के लिए किया जाता है।
- assert का उपयोग डिबगिंग के लिए यह जांचने हेतु किया जाता है कि कोई शर्त सत्य है या नहीं।

5. **What does the finally block ensure in Python code?**

The finally block ensures that the code inside it will execute regardless of whether an exception occurred or not.

पायथन कोड में finally ब्लॉक क्या सुनिश्चित करता है?

finally ब्लॉक सुनिश्चित करता है कि इसमें लिखा कोड, चाहे अपवाद हो या न हो, हर स्थिति में निष्पादित होगा।

6. **Explain the term "exception propagation".**

Exception propagation refers to the process where an exception is passed to the calling function or method until it is handled.

"अपवाद प्रसार" शब्द को समझाएं।

अपवाद प्रसार उस प्रक्रिया को कहते हैं जिसमें अपवाद को तब तक कॉलिंग फंक्शन या मेथड को पास किया जाता है जब तक कि उसे संभाला न जाए।

7. **Define Syntax Error.**

Syntax errors occur when the code violates Python's syntax rules, preventing the program from running.

सिंटैक्स एरर को परिभाषित करें।

सिंटैक्स एरर तब होती है जब कोड पायथन के सिंटैक्स नियमों का उल्लंघन करता है, जिससे प्रोग्राम चल नहीं पाता।

8. **What is the difference between is and == in Python?**

- is checks if two objects refer to the same memory location.
- == checks if the values of two objects are equal.

पायथन में is और == में क्या अंतर है?

- is जांचता है कि क्या दो ऑब्जेक्ट्स एक ही मेमोरी लोकेशन को संदर्भित करते हैं।
- == जांचता है कि क्या दो ऑब्जेक्ट्स के मान समान हैं।

Section B (Short Answer Type Questions - 12 Questions, 1.5 Marks Each)

1. **Write a Python program to handle a ZeroDivisionError exception.**

Python Program for ZeroDivisionError Handling:

```
try:  
    print(10 / 0)  
except ZeroDivisionError:  
    print("Division by zero is not allowed.")
```


2. **What is the difference between a try block and an except block in Python?**

Difference Between try and except: A try block contains code that might raise exceptions. An except block handles those exceptions, preventing abrupt program termination.

3. **Explain the purpose of the finally block in exception handling.**

Purpose of finally Block: The finally block contains code that always executes, regardless of whether an exception occurred, ensuring resource cleanup or critical final steps.

4. **Give two examples of syntax errors in Python and explain how they occur.**

Syntax Errors Examples:

- Missing colon: `if True print("Hello")`.
- Improper indentation: Code lines misaligned. These errors stop program execution.

5. **Explain the raise statement in Python with an example.**

Raise Statement in Python: The raise statement triggers exceptions.

Example:

```
raise ValueError("Custom error message")
```

6. **Differentiate between NameError and TypeError.**

NameError vs TypeError:

NameError: Undefined variable usage.

TypeError: Invalid operation on data types, e.g., `1 + 'a'`.

7. **Write a short note on EOFError with an example.**

EOFError: Occurs when `input()` reaches end-of-file unexpectedly.

Example:

```
try:  
    input()  
except EOFError:  
    print("End of input reached.")
```

8. **What is an IndexError, and when does it occur in Python?**

IndexError: Raised when accessing an invalid sequence index, e.g., `arr = [1, 2]; arr[5]`. It indicates the index is out of range

9. How does Python handle multiple exceptions in a single program? Write an example code.

Handling Multiple Exceptions: Python allows multiple except blocks.

```
try:
    x = int(input())
    print(10 / x)
except ZeroDivisionError:
    print("Cannot divide by zero")
except ValueError:
    print("Invalid number")
```

10. Write the steps involved in the process of exception handling in Python.

Exception Handling Steps:

- Detect error.
- Raise exception.
- Locate matching handler.
- Execute handler or terminate if none found.

11. Why do we need custom exceptions in Python? Write an example program to create and raise a user-defined exception.

Custom Exceptions: Custom exceptions handle specific cases.

```
class CustomError(Exception):
    pass

raise CustomError("Custom exception raised")
```

12. Explain the term "EOF (End of File)".

EOF (End of File): EOF indicates the end of file or input stream, e.g., reading beyond available data in a file.

Section C (Long Answer Type Questions - 3 Questions, 3 Marks Each)

1. Write a Python program to demonstrate the use of try, except, and finally blocks to handle exceptions.

```
try:
    num = int(input("Enter a number: "))
    print(f"The number is {num}")
except ValueError:
    print("Invalid input! Please enter a number.")
```

```
finally:
```

```
    print("Execution completed.")
```

This program ensures smooth execution regardless of user input by catching exceptions and executing the finally block for cleanup.

2. Explain the types of Python exceptions (built-in and user-defined) with examples for each.

- **Built-in:** Predefined errors like ValueError (e.g., `int("abc")`), FileNotFoundError (e.g., opening a non-existent file).
- **User-defined:** Custom exceptions defined by the programmer using a class.

Example:

```
class CustomError(Exception):
```

```
    pass
```

```
try:
```

```
    raise CustomError("This is a custom error!")
```

```
except CustomError as e:
```

```
    print(e)
```

3. Design a Python program that reads a file and handles the following exceptions:

i. FileNotFoundError ii. ValueError

```
try:
```

```
    with open("sample.txt", "r") as file:
```

```
        data = int(file.read())
```

```
except FileNotFoundError:
```

```
    print("File not found!")
```

```
except ValueError:
```

```
    print("File contains non-integer data.")
```

This program reads a file, catching specific errors like missing files or invalid data.

I. Definition of Exceptions

Exceptions in Python are errors that occur during the execution of a program, disrupting its normal flow. Unlike syntax errors detected during code parsing, exceptions are runtime issues such as division by zero or file not found. Python categorizes these into built-in exceptions (e.g., `ZeroDivisionError`, `ValueError`) and user-defined exceptions.

II. Importance of Exception Handling

Exception handling ensures the program continues running smoothly even when unexpected errors occur. Without it, unhandled exceptions can cause abrupt termination, leading to poor user experience and potential data loss.

III. Role of try, except, else, and finally Clauses

- **try Block:** Contains code that may raise exceptions.
- **except Block:** Captures and handles specific exceptions raised in the try block.
- **else Block:** Executes code if no exception occurs in the try block.
- **finally Block:** Always executes, regardless of exceptions, for cleanup operations like closing files.

IV. Example Program

python

Copy code

try:

```
num = int(input("Enter a number: "))
result = 10 / num
```

except ZeroDivisionError:

```
print("Error: Division by zero is not allowed.")
```

except ValueError:

```
print("Error: Invalid input, please enter a number.")
```

else:

```
print("Result:", result)
```

finally:

```
print("Execution complete.")
```

Raising and Catching User-Defined Exceptions

Python allows developers to create custom exceptions to handle domain-specific errors.

Syntax for Raising User-Defined Exceptions

python

Copy code

```
class CustomError(Exception):
```

```
    pass
```

try:

```
raise CustomError("This is a custom exception.")
```

except CustomError as e:

```
print(e)
```

Benefits

1. Enhances clarity by defining precise error-handling mechanisms.
2. Supports debugging by providing meaningful exception messages.
3. Allows integration of domain-specific logic into error handling.

Python Program with Assertion for Debugging

python

Copy code

```
def check_positive(num):
```

```
    assert num > 0, "Number must be positive!"
```

```
    print(f"The number {num} is positive.")
```

try:

```
check_positive(-5)
```

except AssertionError as e:

```
print(e)
```

Explanation: The assert statement ensures conditions are met during runtime, raising an AssertionError if false. This aids in early error detection during debugging.

40

SAMPAT LILER

>> SIR NOTES COLLECTION <<