* Overview of Exceptions or Run Time Errors
* Pre-defined Exceptions
  + ValueError
  + TypeError
  + KeyError
* Raising Exception

1. What is an exception in Python?

a) A syntax error

b) A runtime error

c) A logical error

d) None of the above

Answer: b) A runtime error

1. Which keyword is used to raise an exception explicitly in Python?

a) raise

b) try

c) except

d) else

Answer: a) raise

1. Which of the following is a pre-defined exception in Python?

a) ValueError

b) TypeError

c) KeyError

d) All of the above

Answer: d) All of the above

1. What does the ValueError exception indicate in Python?

a) It is raised when a function is called with an argument of incorrect type

b) It is raised when a key is not found in a dictionary

c) It is raised when a numeric or mathematical operation has an invalid argument

d) None of the above

Answer: c) It is raised when a numeric or mathematical operation has an invalid argument

1. What does the TypeError exception indicate in Python?

a) It is raised when a function is called with an argument of incorrect type

b) It is raised when a key is not found in a dictionary

c) It is raised when a numeric or mathematical operation has an invalid argument

d) None of the above

Answer: a) It is raised when a function is called with an argument of incorrect type

1. What does the KeyError exception indicate in Python?

a) It is raised when a function is called with an argument of incorrect type

b) It is raised when a key is not found in a dictionary

c) It is raised when a numeric or mathematical operation has an invalid argument

d) None of the above

Answer: b) It is raised when a key is not found in a dictionary

7. Which statement is used to handle an exception in Python?

a) if

b) try

c) raise

d) assert

Answer: b) try

8. Which statement is used to define a block of code to be executed if an exception is raised in Python?

a) if

b) try

c) except

d) else

Answer: c) except

9. Can multiple exceptions be handled in a single except block in Python?

a) Yes

b) No

Answer: a) Yes

10. What is the purpose of the finally block in Python exception handling?

a) To define a block of code to be executed if no exception is raised

b) To define a block of code to be executed if an exception is raised

c) To define a block of code to be executed regardless of whether an exception is raised or not

d) None of the above

Answer: c) To define a block of code to be executed regardless of whether an exception is raised or not.