**Exception Notes**

In Python, an exception is an event that occurs during the mention of a program, disrupting its normal flow. Exception typically arises due to errors such as invalid input, division by zero or accessing invalid indices in a list.

1. **Common Exceptions in Python**
2. **ZeroDivision Error:** Raised when dividing by zero. [ print(10/0) ] #zeroDivisionError.
3. **TypeError:** Raised when an operation is applied to an object of an inappropriate type. print(10/0) #TypeError
4. **ValueError:** Raised when a function receives an argument of the right type but an inappropriate value
5. **IndexError:** Raised when trying to access an index outside the range of a list. lst=[1,2,3] print(lst[5]) #IndexError
6. **KeyError:** Raised when trying to access a non-existent key in a dictionary. D={“key”:”value”} Print(d[“no\_key”]) #key error
7. **Handling Exceptions**
8. You can handle exceptions using a try-except block:

try:

x=10/0

except ZeroDivisionError:

print(“You can’t divide by zero!”)

1. **Raising Exceptions**
2. You can raise your own exceptions using the raise keywotd:

x=-1

if x<0:

raise ValueError(“Negative numbers are not allowed!”)

1. **Custom Exceptions**
2. You can define your own exceptions by subclassing the exception class: class MyCustomError(Exception): pass

try: raise MyCustomError(“Something went wrong!”)

except MyCustomError as e:

print(e)