Chapter 12 - Advanced Python 1 Exception Handling in Python

There are many built-in exceptions which are raised
in Python when something goes wrong.

Exceptions in Python can be handled using a try

Statement. The code that handles the exception is written in the except clause. # code which might throw exception as e: Exception print (e) When the exception is handled, the code flow continues without program interruption. We can also specify the exceptions to catch like below: except ZeroDivision Error: # Code except TypeEmor: # code exception # code → All other exceptions are handled here. Kaising Exceptions We can traise sustain exceptions using the raise keyword in python.

	Sometimes we want to run a piece of code when try was successful.
Service .	Sometimes we want to run a piece of code whom
	fry was successful.
	tey:
	# Some code
1	
	except:
1	# Some Code else:
1	Usla transfer to
-	# code -> This is executed only if the try was saccessful
+	Try was Saturatur
+	1 American
1	try with finally
1	Python offers a finally clause which ensures execution
1	Python offers a finally clause which ensures execution of a fixee of code irrespective of the exception.
	Aller omtitans no broad
	try: # 50me code
1	# Some code se
1	Purcht o
1	exapt:
-	# Some code
-	finally:
-	# some code - executed regardless of error!
-	
	ifname_ == '_main_ in Python
	- name - evaluates to the name of the module in Python
	from where the program is ran
-	The program is not
-	To 4 . I la be're your discatly from the command line
-	If the module is being run directly from the command line, thename is set to string "main_"
-	ne name 15 Set to string main
	Thus this behaviour is used to check whether the
	Thus this behaviour is used to check whether the module is run directly or imported to another file.
-	

The global keyword
global keyword is used to modify the variable outside
of the current scope. Enumerate function in Python
The enumerate function adds counter to an iterable and returns it for i, item in list 1: print (i, item)

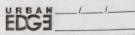
Prints the items of list 1

with index! list comprehension is an elegant way to create lists based on existing lists list 1 = [1, 7, 12, 11, 22] list 2 = [i for item in list 1 if item 78]

EDGE Chapter 10 - Object Oriented Programming Solving a problem by creating objects is one of the most popular approaches in programming. This is called Object oriented programming. This concept focuses on using reusable code.

Implements DRY principle A class is a blueprint for creating objects. Blank => Filled by an student => Application of the Student contains info to Creak a valid Application 4 ⇒ Object instantiation ⇒ Object Contains info to Creak a valid object The syntax of a class looks like this: Class Employee: # methods & variables Classname is written in Pascallase Object An Object is an instantiation of a class. When class is defined, a template (info) is defined. Memory is allocated only after object instantiation. Objects of a given class can invoke the methods available to it without revealing the implementation details to the user.

Abstraction & Encapsulation.



Modelling a problem in OOPs We identify the following in our problem Noun → Class → Employee

Adjective → Attributes → name, age, Salary

Verbs → Methods → getSalary(), increment() Class Attributes An attribute that belongs to the class rather than a particular object. Example: Class Employee:

Company = "Google" → Specific to each class → object instantiation harry = Employer()
harry Company

Employee company = "YouTube"
harry dass attribute An attribute that belongs to the Instance (object) Assuming the class from the previous example: harry name = "Harry"
harry 5alary = "30 K" => Adding instance attributes Note: Instance attributes take preference over class attributes during assignment & retreival harry attributes - 15 attributes present in object?

1 Is attributes present in class?

EDG self parameter
self refers to the instance of the class
It is automatically passed with a function call
from an object harry get Salary () -> here self is harry equivalent to Employee get Salary (here) The function getsalary is defined as: Class Employee:

Company = "Google"

def get Salary (Self):

print ("Salary is not there") Sometimes we need a function that doesn't use the Self parameter we can define a static method like this: @ Stationethod def greet ():

def greet ():

as a static method print ("Hello user") -init_() Constructor _init__() is a special method which is first run as soon as the object is created:
_init__() method is also known as constructor It takes self argument and can also take further arguments

-	
	For Example:
	Class Employee:
	del - init - (self name):
	Self: name = name
	Class Employee: def init (self, name): Self-name = name
	def get Salary (Self):
	harry = Employee ("Harry") Deject can be instantiated using constructor like this!
	harry = Employee ("Harry")
	- Object can be instantiated
	using constructor like this!
124	
-	