## Lambda Function

A lambda is a small anonymous function that can be written in a single line of code.  
Syntax: lambda arguments: expression  
Example:  
add = lambda x, y: x + y  
print(add(5, 3)) # Output: 8  
Lambdas provide a quick way to write functions without using the def keyword.

## Important Python Concepts:

### 1. Function

A function is a block of code that performs a specific task. It's convenient for reusing code.  
Syntax:  
def function\_name(arguments):  
 return value  
Example:  
def greet(name):  
 return f"Hello, {name}!"  
  
print(greet("Rishi")) # Output: Hello, Rishi!

### 2. Conditional Statements (if, elif, else)

These statements check conditions and execute code accordingly.  
Example:  
age = 18  
if age >= 18:  
 print("You're an adult!")  
elif age > 12:  
 print("You're a teenager!")  
else:  
 print("You're a child!"

### 3. Loops (for, while)

For Loop:  
A for loop runs code a predefined number of times.  
Example:  
for i in range(5):  
 print(i) # Output: 0, 1, 2, 3, 4  
  
While Loop:  
A while loop runs as long as a condition is true.  
Example:  
count = 0  
while count < 5:  
 print(count)  
 count += 1

### 4. Exception Handling (try, except)

Exception handling is used when there is a possibility of an error, preventing the program from crashing.  
Example:  
try:  
 result = 10 / 0  
except ZeroDivisionError:  
 print("You can't divide by zero!")

### 5. List Comprehension

A compact way to generate lists.  
Example:  
squares = [x\*\*2 for x in range(10)]  
print(squares) # Output: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

### 6. Modules and Packages

Modules and packages are pre-written code that extend Python's functionality, like adding features to a calculator.  
Importing a Module:  
import math  
print(math.sqrt(16)) # Output: 4.0  
  
Using a Package:  
Packages are collections of modules. Python has many built-in packages.