Functional Programming Lab 01

K M Anisul Islam

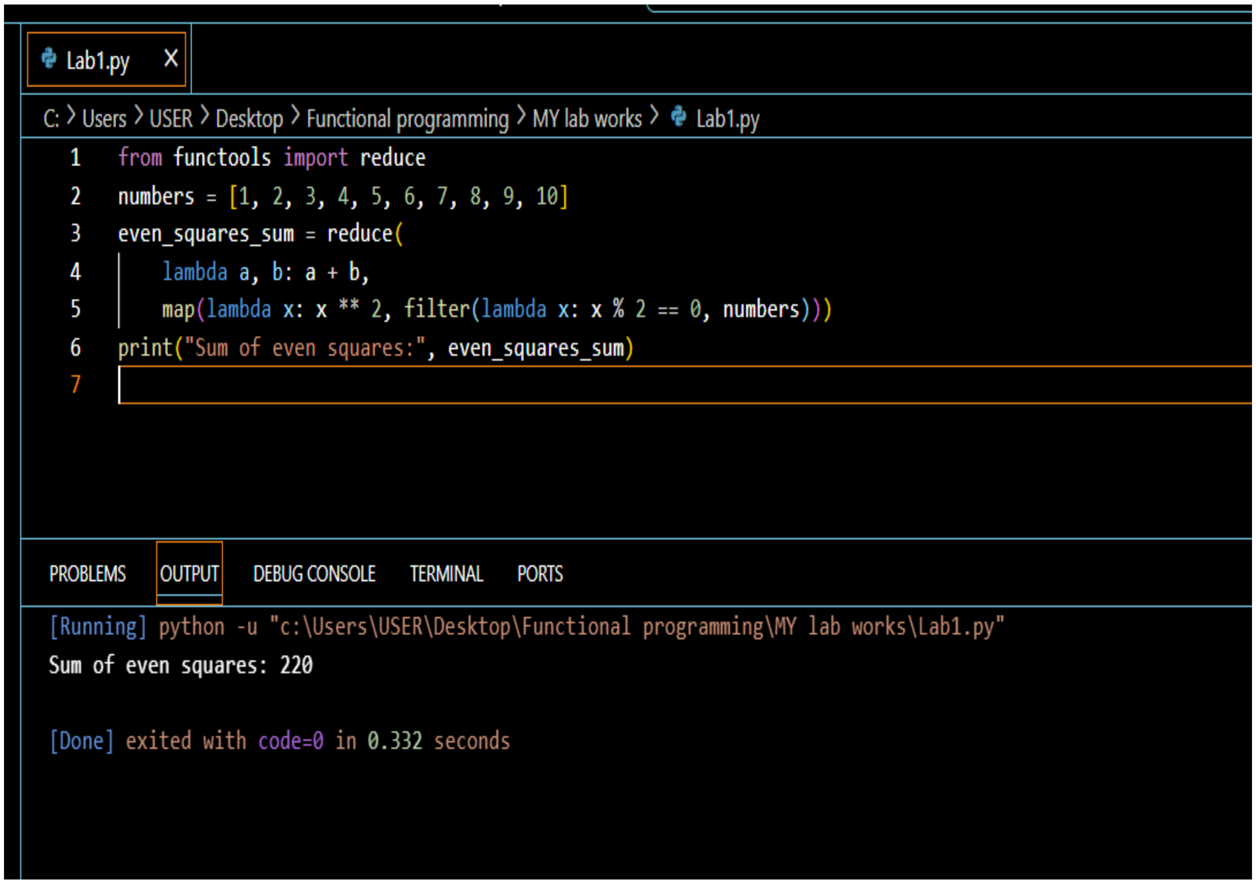
Email: ISLAM.K@stud.satbayev.university

Objective: To study the basics of functional programming in Python, including understanding and applying key concepts such as pure functions, data immutability, lambda functions, and the use of map, filter, and reduce

Individual Task:

Task 1: Sum of Even Squares

Filter even numbers, square them, and find their sum.


Lab1.py
C: > Users > USER > Desktop > Functional programming > MY lab works > Lab1.py
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```python
from functools import reduce
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_squares_sum = reduce(
    lambda a, b: a + b,
    map(lambda x: x ** 2, filter(lambda x: x % 2 == 0, numbers)))
print("Sum of even squares:", even_squares_sum)
```

```
[Running] python -u "c:\Users\USER\Desktop\Functional programming\MY lab works\Lab1.py"
Sum of even squares: 220

[Done] exited with code=0 in 0.332 seconds
```

1. What is functional programming, and how does it differ from the imperative style of programming?
   # It is an Imperative programming focuses on how to perform tasks step by step by changing program state.It treats computation as evaluation of mathematical functions and avoids changing state or mutable data. Imperative programming focuses on how to perform tasks step by step by changing program state.
2. What are the advantages of using the functional approach in programming?
   #It provides easier debugging, better modularity, and improved readability. It also makes programs easier to test since functions are independent and pure.
3. What is a pure function? Provide an example.
   # A pure function always produces the same output for the same input and has no side effects.
   Such as,
   def square(x): return x * x

4. What are lambda functions in Python, and how are they used?
   # Lambda functions are small anonymous functions defined using the 'lambda' keyword.
5. How do map, filter, and reduce functions work? Provide examples
   # map(func, iterable) applies a function to all elements.
   filter(func, iterable) keeps elements where func returns True.
   reduce(func, iterable) combines elements using a binary function.
   Such as: reduce(lambda a,b: a+b, [1,2,3]) → 6

6. How can recursion be implemented in Python, and what are its limitations.
   # Recursion is when a function calls itself to solve smaller instances of a problem. Its limitation is Python’s maximum recursion depth
7. What are the methods for optimizing recursive functions in Python?
   # Memoization (storing previous results) and tail recursion optimization are common methods. Python supports memoization using functools.lru_cache.
8. How can functional programming improve code readability and maintainability?
   # It makes code cleaner and easier because each function performs one clear task without modifying global state or variables.
9. What is the difference between mutable and immutable data types in Python? Provide examples.
   Mutable data types can be changed after creation (lists, dictionaries). Immutable types cannot be modified after creation (tuples, strings).
10. Can functional programs be efficient in terms of performance? Explain your answer
    Yes, However, they may be slower than imperative code in some cases due to recursion and immutable data. Optimizations and built-in functional tools help mitigate this.