

## Lab3a

```
[Preview] README.md  lab3a.py M X  lab3b.py M  lab3c.py M  README.md  lab3d.py M  lab3e.py M  lab3f.py M  lab3g.py M  Python + - [ ] ... [ ]
lab3a.py > ...
1  # Add comments before you do anything else.
2
3  #!/usr/bin/env python3
4  # Author: Soroush Bastani sbastani1
5  # Date: 2025-10-02
6  # Purpose: Create Simple Functions.
7  # Usage: ./lab3a.py
8
9  # TO DO 1: Add the docstring
10 # @Function definition: Checks if any number in a list is even.
11 # @param: a_list - A list of integer values.
12 # @return: bool - True if an even number is found, False otherwise.
13
14 # TO DO 2: define the function with name 'is_even'.
15 def is_even(a_list):
16     for number in a_list:
17         if number % 2 == 0:
18             return True
19     return False
20
21 # TO DO 3: Call the function 'is_even'.
22 my_numbers = [1, 3, 5, 8, 9, 11]
23 result = is_even(my_numbers)
24 print(result)
```

```
TERMINAL
/home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3a.py
@Soroush-Bastani → /workspaces/lab-3-Soroush-Bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3a.py
True
@Soroush-Bastani → /workspaces/lab-3-Soroush-Bastani (main) $
```

## Lab3b

```
[Preview] README.md lab3a.py M lab3b.py M X lab3c.py M README.md lab3d.py M lab3e.py M lab3f.py M lab3g.py M
```

```
lab3b.py > ...
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python3
4 # Author: Soroush Bastani sbastani1
5 # Date: 2025-10-02
6 # Purpose: Create Some Complex Functions.
7 # Usage: ./lab3b.py
8
9 # TO DO 1: Add the docstring
10 # @Function definition: Filters a list of integers and returns a new list containing only the even numbers.
11 # @param: num_list - A list of integer values.
12 # @return: list - A new list containing only the even numbers from the original list.
13
14 # TO DO 2: Create the function.
15 def even_numbers(num_list):
16     even_list = []
17     for number in num_list:
18         if number % 2 == 0:
19             even_list.append(number)
20     return even_list
21
22 # TO DO 3: Call the function.
23 original_list = [10, 21, 34, 45, 58, 69, 72, 83]
24 filtered_list = even_numbers(original_list)
25 print(filtered_list)
```

```
TERMINAL Python + v | | |
/home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3b.py
@Soroush-Bastani → /workspaces/lab-3-Soroush-Bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3b.py
[10, 34, 58, 72]
@Soroush-Bastani → /workspaces/lab-3-Soroush-Bastani (main) $
```

## Lab3c



The image shows a code editor with a file explorer at the top displaying several files: [Preview] README.md, lab3a.py M, lab3b.py M, lab3c.py M (selected), README.md, lab3d.py M, lab3e.py M, lab3f.py M, and lab3g.py M. The main editor area shows the content of lab3c.py, which includes comments and a Python script. The script defines a 'sum' function and a 'main' function that takes user input and prints the result. The terminal on the right shows the command to run the script, the user input for two numbers (6 and 9), and the output 'The result is: 15'.

```
1 # Add comments before you do anything else.
2 #!/usr/bin/env python3
3 # Author: Soroush Bastani sbastani1
4 # Date: 2025-10-02
5 # Purpose: use the main Function as entry point.
6 # Usage: ./lab3c.py
7
8 # TO DO 1: Write a function named `sum` that takes two numbers and returns their sum.
9 def sum(num1, num2):
10     return num1 + num2
11
12 # TO DO 2: Write a `main()` function that prompts the user for two numbers, calls `sum()`, and prints the result.
13 def main():
14     input1 = input("Enter the first number: ")
15     input2 = input("Enter the second number: ")
16     num1 = int(input1)
17     num2 = int(input2)
18     result = sum(num1, num2)
19     print(f"The result is: {result}")
20 if __name__ == "__main__":
21     main()
22
```

TERMINAL

```
/home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3c.py
@Soroush-Bastani → /workspaces/lab-3-Soroush-Bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3c.py
Enter the first number: 6
Enter the second number: 9
The result is: 15
@Soroush-Bastani → /workspaces/lab-3-Soroush-Bastani (main) $
```

## Lab3d

[Preview] README.mdlab3a.py Mlab3b.py Mlab3c.py MREADME.mdlab3d.py M Xlab3e.py Mlab3f.py Mlab3g.py M

lab3d.py > compute

```
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python3
4 # Author: Soroush Bastani sbastani1
5 # Date: 2025-10-02
6 # Purpose: Create the complete calculator function using default parameters and positional parameters
7 # Usage: ./lab3d.py
8
9 # Write a function named compute
10 def compute(num1, num2, operation='+'):
11     if operation == '+':
12         return num1 + num2
13     elif operation == '-':
14         return num1 - num2
15     elif operation == '*':
16         return num1 * num2
17     elif operation == '/':
18         if num2 != 0:
19             return num1 / num2
20         else:
21             return "Error: Division by zero."
22     else:
23         return "Error: Invalid operation."
24
25 # Write a main() function
26 def main():
27     try:
28         user_num1 = int(input("Enter the first number: "))
29         user_num2 = int(input("Enter the second number: "))
30         user_op = input("Choose an operation (+, -, *, /): ")
31
32         result = compute(user_num1, user_num2, user_op)
33         print(result)
34
35     except ValueError:
36         print("Invalid input. Please enter valid integers.")
37
38 # Demonstrate the function with the following calls
39 print(compute(13, 45, '*'))
40 print(compute(13, 45, '/'))
41 print(compute(13, 45, '-'))
42 print(compute(13, 45, '+'))
43 print(compute(13, 45))
44
45 # Call the main function in the conditional statement
46 if __name__ == "__main__":
47     main()
```

Python + ▾ 🗑️ ⋮

TERMINAL

```
/home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3d.py
@Soroush-Bastani → /workspaces/lab-3-Soroush-Bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3d.py
585
0.28888888888888886
-32
58
58
Enter the first number:
```

## Lab3e

[Preview] README.mdlab3a.py Mlab3b.py Mlab3c.py MREADME.mdlab3d.py Mlab3e.py M Xlab3f.py Mlab3g.py M

lab3e.py >...

```
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python3
4 # Author: Soroush Bastani sbastani1
5 # Date: 2025-10-02
6 # Purpose: Modify the calculator program to use keyword parameters.
7 # Usage: ./lab3e.py
8
9 # TO DO 1: Copy the `compute` function from lab3d.py.
10 def compute(num1, num2, operation='+'):
11     if operation == '+':
12         return num1 + num2
13     elif operation == '-':
14         return num1 - num2
15     elif operation == '*':
16         return num1 * num2
17     elif operation == '/':
18         if num2 != 0:
19             return num1 / num2
20         else:
21             return "Error: Division by zero."
22     else:
23         return "Error: Invalid operation."
24
25 # TO DO 2: Demonstrate the function with calls using keyword arguments.
26 print(compute(num1=13, num2=45, operation='*'))
27 print(compute(operation='/', num2=45, num1=13))
28 print(compute(num2=45, num1=13, operation='-'))
29 print(compute(num1=13, num2=45, operation='+'))
30 print(compute(num1=13, num2=45))
```

TERMINALPython + ▢ ⌵ ⌵ ⌵ ⌵  
/home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3e.py  
@Soroush-Bastani → /workspaces/lab-3-Soroush-Bastani (main) \$ /home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3e.py  
585  
0.28888888888888886  
-32  
58  
58  
@Soroush-Bastani → /workspaces/lab-3-Soroush-Bastani (main) \$

# Lab3f

```
[Preview] README.md lab3a.py M lab3b.py M lab3c.py M README.md lab3d.py M lab3e.py M lab3f.py M X lab3g.py M
```

```
lab3f.py > ...
1  # Add comments before you do anything else.
2
3  #!/usr/bin/env python3
4  # Author: Soroush Bastani sbastani1
5  # Date: 2025-10-02
6  # Purpose: Practice variable number of arguments with *args
7  # Usage: ./lab3f.py
8
9  # TO DO 1: Write a function named `get_initials` that uses *args.
10 def get_initials(*names):
11     """
12     Uses *args to accept a variable number of name inputs,
13     and returns a list of the first letter of each name.
14     """
15     initials = []
16     for name in names:
17         # Ensure the name is not an empty string before getting the initial
18         if name:
19             initials.append(name[0])
20     return initials
21
22 # TO DO 2: Call the function and print the results as per instructions.
23 # Call the get_initials() function with the names "Samuel", "Ravi", "Chen", "Fatima".
24 student_initials = get_initials("Samuel", "Ravi", "Chen", "Fatima")
25
26 # Print the result.
27 print(student_initials)
28
29 # The following lines demonstrate the function with the other examples from the lab description.
30 print(get_initials("Emma", "Maija", "Sophia"))
31 print(get_initials("John"))
32 print(get_initials("Olivia", "Ravi", "Chen", "Fatima"))
```

```
TERMINAL
/home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3f.py
@soroush-Bastani → /workspaces/lab-3-Soroush-Bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3f.py
['S', 'R', 'C', 'F']
['E', 'M', 'S']
['J']
['O', 'R', 'C', 'F']
@soroush-Bastani → /workspaces/lab-3-Soroush-Bastani (main) $
```

# Lab3g

```
[Preview] README.md lab3a.py M lab3b.py M lab3c.py M README.md lab3d.py M lab3e.py M lab3f.py M lab3g.py M X
```

```
lab3g.py > ...
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python3
4 # Author: Soroush Bastani sbastani1
5 # Date: 2025-10-02
6 # Purpose: Practice map, filter and lambda expressions.
7 # Usage: ./lab3g.py
8
9 # Create a variable numbers containing numbers from 2 to 10.
10 numbers = list(range(2, 11))
11
12 # Square all elements of this list using map and a lambda function.
13 # Store the result back in numbers.
14 numbers = list(map(lambda x: x ** 2, numbers))
15
16 # Print the variable numbers.
17 print(numbers)
18
19 # Create a new variable named divisible_by_2.
20 # Use filter and a lambda function to select all numbers from numbers that are divisible by 2.
21 divisible_by_2 = list(filter(lambda x: x % 2 == 0, numbers))
22
23 # Print the variable divisible_by_2.
24 print(divisible_by_2)
25
```

```
TERMINAL
/home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3g.py
@soroush-bastani → /workspaces/lab-3-Soroush-Bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3g.py
[4, 9, 16, 25, 36, 49, 64, 81, 100]
[4, 16, 36, 64, 100]
@soroush-bastani → /workspaces/lab-3-Soroush-Bastani (main) $
```