

Lab3a

The screenshot shows a code editor interface with a dark theme. On the left is the code editor pane containing the file `lab3a.py`. The code is as follows:

```
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)
```

On the right is a terminal pane titled "TERMINAL". It shows the command `/home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3a.py` being run, followed by the output `True`.

Lab3b

The screenshot shows a Jupyter Notebook interface with several tabs at the top: [Preview] README.md, lab3a.py M, lab3b.py M (which is the active tab), lab3c.py M, README.md, lab3d.py M, lab3e.py M, lab3f.py M, lab3g.py M, and On the left, the code for lab3b.py is displayed:

```
lab3b.py > ...
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python3
4 # Author: Soroush Bastani
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)
```

On the right, the terminal window shows the output of running the script:

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

Lab3c

```
[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 D v ...]
lab3c.py > ...
1 # Add comments before you do anything else.
2 #!/usr/bin/env python
3 # Author: Soroush Bastani sbastani
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.md lab3a.py M lab3b.py M lab3c.py M README.md lab3d.py M lab3e.py M lab3f.py M lab3g.py M ⌂ ⌂ ⌂ ...

```
lab3d.py > ⌂ compute
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python3
4 # Author: Soroush Bastani sbastani
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()
```

TERMINAL

```
/home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3d.py
/home/codespace/.python/current/bin/python /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

The screenshot shows a code editor interface with a dark theme. On the left, there is a navigation bar with file icons and names: READMEmd, lab3a.py M, lab3b.py M, lab3c.py M, README md, lab3d.py M, lab3e.py M (highlighted), lab3f.py M, lab3g.py M, and In the center, the main code editor area contains the following Python script:

```
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 calcualtor 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))
```

To the right of the code editor is a terminal window titled "TERMINAL". It shows the command `/home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3e.py` being run, followed by the output:

```
@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

The screenshot shows a code editor interface with a terminal window on the right. The code editor has tabs for README.md, lab3a.py, lab3b.py, lab3c.py, README.md, lab3d.py, lab3e.py, lab3f.py (selected), lab3g.py, and The terminal window shows the output of running the lab3f.py script, which prints a list of initials.

```
#!/usr/bin/env python3
# Author: Soroush Bastani1
# Date: 2025-10-02
# Purpose: Practice variable number of arguments with *args
# Usage: ./lab3f.py

# TO DO 1: Write a function named `get_initials` that uses *args.
def get_initials(*names):
    """
    Uses *args to accept a variable number of name inputs,
    and returns a list of the first letter of each name.
    """
    initials = []
    for name in names:
        # Ensure the name is not an empty string before getting the initial
        if name:
            initials.append(name[0])
    return initials

# TO DO 2: Call the function and print the results as per instructions.
# Call the get_initials() function with the names "Samuel", "Ravi", "Chen", "Fatima".
student_initials = get_initials("Samuel", "Ravi", "Chen", "Fatima")

# Print the result.
print(student_initials)

# The following lines demonstrate the function with the other examples from the lab description.
print(get_initials("Emma", "Maija", "Sophia"))
print(get_initials("John"))
print(get_initials("Olivia", "Ravi", "Chen", "Fatima"))
```

```
/home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3f.py
@Soroush-Bastani → /workspaces/lab-3-Soroush-Bastani (main) $ ./home/codespace/.pyt
on/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

The screenshot shows a Jupyter Notebook interface with the following components:

- File Bar:** Contains links to README.md, lab3a.py, lab3b.py, lab3c.py, README.md, lab3d.py, lab3e.py, lab3f.py, lab3g.py (selected), and a Help icon.
- Code Cell:** Displays the content of `lab3g.py`. The code uses map, filter, and lambda functions to manipulate a list of numbers from 2 to 10, squaring them and then filtering for even numbers.
- Terminal:** Shows the command `/home/codespace/.python/current/bin/python /workspaces/lab-3-Soroush-Bastani/lab3g.py` being run, followed by the output: `[4, 16, 25, 36, 49, 64, 81, 100]` and `[4, 16, 36, 64, 100]`.

```
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, 16, 25, 36, 49, 64, 81, 100]
[4, 16, 36, 64, 100]
@Soroush-Bastani → /workspaces/lab-3-Soroush-Bastani (main) $
```