

LAB 07

Question 07

```
def get_array_from_user(size):  
    arr = []  
    for i in range(size):  
        val = float(input(f"Enter element  
{i+1}: "))  
        arr.append(val)  
    return arr
```

```
def scalar_sum(array):  
    return sum(array)
```

```
def vector_sum(array1, array2):  
    return [a + b for a, b in zip(array1,  
array2)]
```

```
def vector_product(array1, array2):  
    return [a * b for a, b in zip(array1,  
array2)]
```

```
def scalar_product(array1, array2):  
    return sum(a * b for a, b in zip(array1,  
array2))  
  
# Get the size of the arrays from the user  
size = int(input("Enter the size of the  
arrays: "))  
  
# Get the elements for the first array  
print("Enter elements for the first array:")  
array1 = get_array_from_user(size)  
  
# Get the elements for the second array  
print("Enter elements for the second  
array:")  
array2 = get_array_from_user(size)  
  
# Calculate Scalar Sum  
scalar_sum_result = scalar_sum(array1)  
print("Scalar Sum:", scalar_sum_result)
```

```
# Calculate Vector Sum
vector_sum_result = vector_sum(array1,
array2)
print("Vector Sum:", vector_sum_result)
```

```
# Calculate Vector Product
vector_product_result =
vector_product(array1, array2)
print("Vector Product:",
vector_product_result)
```

```
# Calculate Scalar Product
scalar_product_result =
scalar_product(array1, array2)
print("Scalar Product:",
scalar_product_result)
```

Question 08

```
class Animal:
    def animal_method(self):
        print("I am an Animal")

class Dog(Animal):
    def dog_method(self):
        print("I have four legs")

def main():
    # Create a Dog object
    dog_obj = Dog()

    # Call methods from both classes
    dog_obj.animal_method()
    dog_obj.dog_method()

if __name__ == "__main__":
    main()
```