

↑

Essentials of Data Science  
Laboratory - 2304102L

Pin

<

🔍

Search course

ctrl + k

1. Practical 1

^

1.1. Practice Lab Assignment

▼

1.2. Lab Assignment

▼

2. Practical 2

^

2.1. Practice Lab Assignment

▼

2.2. Lab Assignment

▼

3. Practical 3

^

3.1. Practice Lab Assignment

▼

3.2. Lab Assignment

▼

4. Practical 4

▼

5. Practical 5

▼

# Practical 1

## About this unit

Practical 1

### Practice Lab Assignment

Unit • 100% completed



### Lab Assignment

Unit • 100% completed



## 1.1.1. Calculate Momentum

12:40



Write a program that accepts the mass of an object (in kilograms) and its velocity (in meters per second), then calculates and displays the momentum of the object. The momentum  $p$  is calculated using the formula:

$$p = m \times v$$

where:

$m$  is the mass of the object (in kilograms).

$v$  is the velocity of the object (in meters per second).

**Input Format:**

A single floating-point number representing the mass of the object in kilograms.

A single floating-point number representing the velocity of the object in meters per second.

**Output Format:**

The output will display calculated momentum with appropriate units (kgm/s) (rounded up to 2 decimal places).

Sample Test Cases



calculate...



Submit

```
1 m=float(input())
2 v=float(input())
3 p=m*v
4 print('%0.2f'%p,end='')
5 print("kgm/s")
6
7
```

Terminal

Test cases



## 1.1.2. Conditional Calculation Based on the Number of Digits

20:50

Write a Python program that accepts an integer  $n$  as input. Depending on the number of digits in  $n$ .

**Constraints:** $1 \leq n \leq 999$ **Input Format:**

The input consists of a single integer  $n$ .

**Output Format:**

If  $n$  is a single-digit number, print its square.

If  $n$  is a two-digit number, print its square root (rounded to two decimal places).

If  $n$  is a three-digit number, print its cube root (rounded to two decimal places).

Else print "Invalid".

Sample Test Cases

+

condition...

Submit

```
1 n=int(input())
2 if(n>=0 and n<=9):
3     print(n*n)
4
5 elif(n>=10 and n<=99):
6     p=n**0.5
7     print("%.2f"%p)
8
9 elif(n>=100 and n<=999):
10    r=n**(1/3)
11    print("%.2f"%r)
12
13 else:
14    print("Invalid")
```

Terminal

Test cases

### 1.1.3. Age and Salary Calculation

00:54

Write a Python program that reads the birth date and salary of employees.

#### Input Format:

The input consists of:

A string representing the birth date of the employee in the format *DD – MM – YYYY*.

A floating-point number representing the salary of the employee in rupees.

#### Output Format:

The output should include:

The age of the employee.

The salary of the employee in dollars.

#### Note:

1INR=0.012USD

Sample Test Cases

+

birthDate...

Submit

```
1 from datetime import datetime
2
3 def calculate_age(birthdate):
4     date_object = datetime.strptime(birthdate, "%d-%m-%Y")
5     today = datetime.today()
6     age = today.year - date_object.year
7     if (today.month, today.day) < (date_object.month, date_object.day):
8         age -= 1
9     return age
10
11
12
13 def convert_salary_to_dollars(salary_in_rupees):
14     return salary_in_rupees * 0.012
15
16 birthdate = input()
17 salary_in_rupees = float(input())
18 age = calculate_age(birthdate)
19 salary_in_dollars = convert_salary_to_dollars(salary_in_rupees)
20 print(f"Age: {age}")
21 print(f"Salary in dollars: {salary_in_dollars:.2f}")
22
```

Terminal

Test cases





#### 1.1.4. Reverse a Number

05:12

You are given an integer number. Your task is to reverse the digits of the number and print the reversed number.

##### Input Format

The input is an integer.

##### Output Format

Print a single integer which is the reversed number.

Sample Test Cases

+

reverseN...

Submit

```
1 number = int(input())
2 reverse=0
3 while number != 0:
4     digit = number % 10
5     reverse = reverse*10 + digit
6     number = number//10
7
8
9 print(reverse)
10
```

Terminal

Test cases

### 1.1.5. Multiplication Table

Write a program that takes an integer as input and prints the multiplication table for that integer from 1 to 10.

#### Input Format:

The first line of input contains an integer that represents the number for which the multiplication table is to be printed.

#### Output Format:

Print the multiplication table for the given number .

Sample Test Cases



### multiplica...

Submit

```
1 i=int(input())
2 n=1
3 while n<=10:
4     print(i,"x",n,"=",i*n)
5     n=n+1
```

Terminal

Test cases

### 1.2.1. Pass or Fail

#### Pass or Fail

Write a Python program that accepts the number of courses and the marks of a student in those courses.

The grade is determined based on the aggregate percentage:

- If the aggregate percentage is greater than 75, the grade is Distinction.
- If the aggregate percentage is greater than or equal to 60 but less than 75, the grade is First Division.
- If the aggregate percentage is greater than or equal to 50 but less than 60, the grade is Second Division.
- If the aggregate percentage is greater than or equal to 40 but less than 50, the grade is Third Division.

#### Input Format:

The first input will be an integer  $n$ , the number of courses.

The second input will be  $n$  integers representing the marks of the student in each of the  $n$  courses, separated by a space.

#### Output Format:

If the student passes all courses:

- Print the aggregate percentage (rounded to two decimal places).
- Print the grade based on the aggregate percentage.

If the student fails any course (marks < 40 in any course), print:

Sample Test Cases

```
passorFa...
1 n = int(input())
2 marks = list(map(int, input().split()))
3
4 if any(mark < 40 for mark in marks):
5     print("Fail")
6 else:
7     totalmarks = sum(marks)
8     aggregate = (totalmarks/(n*100))*100
9     print(f"Aggregate Percentage: {aggregate:.2f}")
10
11 if aggregate > 75:
12     print("Grade: Distinction")
13 elif aggregate > 60 and aggregate < 75:
14     print("Grade: First Division")
15 elif aggregate > 50 and aggregate < 60:
16     print("Grade: Second Division")
17 elif aggregate > 40 and aggregate < 50:
18     print("Grade: Third Division")
19 else:
20     print("Fail")
```

Terminal Test cases

## 1.2.2. Fibonacci series using Recursive Function

04:13

Write a Python program to find the Fibonacci series of a given number of terms using recursive function calls.

### Expected Output-1:

Enter terms for Fibonacci series: 5

0 1 1 2 3

### Expected Output-2:

Enter terms for Fibonacci series: 9

0 1 1 2 3 5 8 13 21

### Instructions:

- Your input and output must follow the input and output layout mentioned in the visible sample test case.
- Hidden test cases will only pass when users' input and output match the expected input and output.

Sample Test Cases

+

```
fib.py
1 def fib(n):
2     if n <= 0:
3         return 0
4     elif n == 1:
5         return 1
6     else:
7         return fib(n - 1) + fib(n - 2)
8
9
10
11
12
13
14
15
16
17
18 n=int(input("Enter terms for Fibonacci series: "))
19 for i in range (n):
20     print(fib(i),end=" ")

Terminal Test cases
```



### 1.2.3. Pattern - 1

57.00

Pattern - 1

Write a Python program to print a pattern of asterisks in the form of a right-angled triangle.

#### Input Format:

The input is an integer, representing the number of rows in the pattern.

#### Output Format

The output should display the pattern of asterisks (\*), with each row containing an increasing number of asterisks.

#### Note:

Refer to the displayed test cases for the sample pattern.

Sample Test Cases

rightangl...

Submit

Debugger

```
1  def print_triangle_pattern(rows):
2      i = 1
3      while i <= rows:
4          print("*" * i)
5          i += 1
6      rows = int(input(""))
7      print_triangle_pattern(rows)
8
```

#### 1.2.4. Pattern - 2

17:34

Write a Python program to print a right-angled triangle pattern of numbers.

##### Input Format:

The input is an integer, representing the number of rows in the pattern.

##### Output Format:

The output should display the pattern of numbers, with each row containing increasing numbers starting from 1 up to the row number.

##### Note:

Refer to the displayed test cases for the sample pattern.

Sample Test Cases



numberP...

Submit

1 def print\_number\_pattern(rows):

2 for i in range(1, rows + 1):

3 print(" ".join(map(str, range(1, i + 1))) + "\n")

4

5 rows = int(input().strip())

6 print\_number\_pattern(rows)

Explorer

Terminal

Test cases

