



Date :

Experiment No.

1 Develop a program to read the student's details, like name, USN and marks for three subjects. Displaying the student details, total marks and percentage with suitable messages.

Input :

studName = input ("Enter student name")

studUSN = input ("Enter student USN")

sm1 = int (input ("Enter subject 1 marks"))

sm2 = int (input ("Enter subject 2 marks"))

sm3 = int (input ("Enter Subject 3 marks"))

total = sm1 + sm2 + sm3

per = (total / 300) * 100

print ("Student name :- ", studName)

print ("Student USN :- ", studUSN)

print ("Total marks secured :- ", total)

print ("percentage scored :- ", per)

if per > 80:

 print ("Grade scored is A")

if per > 70:

 print ("Grade Scored is B")

if per > 60:

 print ("Grade Scored is C")

if per > 45:

 print ("Grade Scored is D")

else:

 print ("Grade Scored is F-Fail")

Output :

Enter student name Irman

Enter student USN HMCE23EE107

Enter subject 1 marks 100

Enter subject 2 marks 100

Enter Subject 3 marks 100

Student name :- Irman

Student USN :- 4MC23EE107

Total marks Secured :- 300

Percentage Scored :- 100.0

Grade Scored is A



Date :

Experiment No.

2) Develop a program to generate fibonacci series of length (N).
Read N from the console.

Input:

`n = int(input("Enter fibonacci sequence number"))`

`a = 0`

`b = 1`

`print(a)`

`for i in range(n-1):`

`a, b = b, a+b`

`print(a)`

Output:

Enter fibonacci sequence numbers 5

0
1
1
2
3



Date :

Experiment No.

3. Write a program to guess the number, calculate the factorial of a number. Develop a program to compute binomial coefficient.

Input:

```
def fact(n):
```

```
i=1
```

```
res=1
```

```
while(i<=n):
```

```
    res = res * i
```

```
    i = i + 1
```

```
return res
```

```
N = int(input("Enter value of N"))
```

```
R = int(input("Enter value of R"))
```

```
BRES = fact(N) / (fact(R) * fact(N-R))
```

```
print("The binomial coefficient for the given N=%d and for  
R=%d is %d", % (N, R, BRES))
```

Output:

Enter value of N 5

Enter value of R 3

The binomial coefficient for the given $N=5$ and for $R=3$ is 10



4. Write a program to guess the number using randint() function
(ask player to guess 6 times)

Input :

```
import random
SecretNumber = random.randint(1,20)
print ("I am thinking of a number between 1 and 20")
for guessesTaken in range(1,7):
    print ("Take a guess")
    guess = int (input())
    if guess < SecretNumber:
        print ('Your guess is too low')
    elif guess > SecretNumber:
        print ('Your guess is too high')
    else:
        break
    if guess == SecretNumber:
        print ('Good job! you guessed my number in ' + str (guessesTaken)
              + ' guesses!')
    else:
        print ('Nope, The number I was thinking of was ' + str
              (SecretNumber))
```

output:

I am thinking of a number between 1 and 20

Take a guess

13

Your guess is too high

Take a guess

12

Good job! You guessed my number in 2 guesses!



Date :

Experiment No.

5. Write a program to get random output from magic 8 ball game.

Input:

```
import random
messages = ['It is certain',
'It is decidedly so',
'Yes definitely',
'Reply "lazy to you again"',
'Ask again later',
'Concentrate and ask again',
'My reply is no',
'Out look not so good',
'Very doubtful.']
print(messages[random.randint(0, len(messages)-1)])
```

output 1:
outlook not so good

output 2:
Yes definitely

output 3:
It is decidedly so,



6) Read a multi-digit number (as chars) from the console. Develop a program to print the frequency of each digit with suitable message.

Input:

```
n = input("Enter a multi-digit number: ")
```

```
freq = {}
```

```
for i in range(len(n)):
```

```
    if n[i] in freq:
```

```
        freq[n[i]] += 1
```

```
    else
```

```
        freq[n[i]] = 1
```

```
for key, value in freq.items():
```

```
    print(f"The frequency of digit {key} is {value}")
```

Output:

Enter a multi-digit number: 12234

The frequency of digit 1 is 1

The frequency of digit 2 is 2

The frequency of digit 3 is 1

The frequency of digit 4 is 1

Date :

Experiment No.

7)

Read N numbers from the console and create a list. Develop a program to print mean, variance and standard deviation with suitable messages.

import math

$N = \text{int}(\text{input}(\text{"Enter the number of elements: "}))$

numbers = []

for i in range $[N]$:

 num = float(input("Enter number " + str(i + 1) + ":"))

 numbers.append(num)

mean = sum(numbers) / N

variance = sum((X - mean) ** 2 for X in numbers) / N

std_deviation = math.sqrt(variance)

print(f"Mean of the list of numbers is : {mean}")

print(f"Variance : {variance}")

print(f"Standard Deviation : {std_deviation}")

Output:

Enter the number of elements : 6

Enter Number 1 : 3.4

Enter Number 2 : 5

Enter Number 3 : 5.3

Enter Number 4 : 7

Enter Number 5 : 2

Enter Number 6 : 8.3

Mean of the list [3.4, 5.0, 5.3, 7.0, 2.0, 8.3] is : 5.1667

Variance : 4.39556

Standard Deviation : 2.096558025802185

Date :

Experiment No.

8)

Write a program to search an element using Linear search.

Input:

```
num = int(input("Enter the number of elements"))
list = []
for i in range(0, num):
    ele = int(input("Enter the elements"))
    list.append(ele)
print(list)
key = int(input("Enter the key to search"))
for i in range(0, len(list)):
    if key == list[i]:
        print("Element found at index", i)
        break
else:
    print("Element not found")
```

output :

Enter the number of elements 5

Enter the elements 1

Enter the elements 2

Enter the elements 3

Enter the elements 4

Enter the elements 5

[1, 2, 3, 4, 5] search 3

Enter the key to search 3

Element found at index 0



Date :

Experiment No.

②

Write a program to search an element using binary search.

```
def binary_search(arr, n):
```

```
    low = 0
```

```
    high = len(arr) - 1
```

```
    mid = 0
```

```
    while low <= high:
```

```
        mid = (high + low) // 2
```

```
        if arr[mid] < n:
```

```
            low = mid + 1
```

```
        elif arr[mid] > n:
```

```
            high = mid - 1
```

```
        else:
```

```
            return mid
```

```
    return -1
```

```
arr = [2, 3, 4, 10, 10]
```

```
n = 2
```

```
result = binary_search(arr, n)
```

```
if result != -1:
```

```
    print("Element is present at index", result)
```

```
else:
```

```
    print("Element is not present in array")
```

output:

Element is present at index 0



10) Write an object oriented python program to create two time objects, currentTime, which contains the current time, and breadTime, which contains the amount of time it takes for a bread maker to make bread. Then we'll use addTime to figure out when the bread will be done. Use the printTime function to display the time when the bread will be done by the bread.

Input :

class Time :

```
def __init__(self, hour, minute, second):  
    self.hour = hour  
    self.minute = minute  
    self.second = second  
def __add__(self, other):  
    seconds = self.second + other.second  
    minutes = self.minute + other.minute + seconds // 60  
    hours = self.hour + other.hour + minutes // 60  
    def __str__(self):  
        seconds %= 60  
        minutes %= 60  
        return Time(hours, minutes, seconds)  
    return f'{self.hour:02}:{self.minute:02}:{self.second:02}'
```

currentTime = Time(12, 0, 0)

breadTime = Time(1, 3, 0)

doneTime = currentTime + breadTime

print(f"The bread will be done at {doneTime}")

output:

The bread will be done at 13:00:00



11) Develop a program that uses class Student which prompts the user to enter marks in three subjects and calculates total marks, percentage and displays the score card details. [Hint: Use list to store the marks in three subjects and total marks. Use init() method to initialize name USN and the lists to store marks and total, use getMarks() method to read marks onto the list, and display() method to display the score card details.]

Input:

class Student :

```
def __init__(self, name, usn):  
    self.name = name  
    self.usn = usn  
    self.marks = []  
    self.total = 0  
def getMarks(self):  
    for i in range(3):  
        mark = int(input("Enter marks in subject " + str(i+1) + ":"))  
        self.marks.append(mark)  
        self.total += mark  
def display(self):  
    print("Student Name : ", self.name)  
    print("USN : ", self.usn)  
    print("Marks : ", self.marks)  
    print("Total : ", self.total)  
    percentage = (self.total / 300) * 100  
    print("Percentage : ", percentage)  
    if percentage >= 75:  
        print("Grade : A")  
    elif percentage >= 60 and percentage < 75:  
        print("Grade : B")  
    else percentage >= 50 and percentage < 60:  
        print("Grade : C")
```

Date :

Experiment No.

else :

print ("Grade : F")

student = Student (input ("Enter student name : ")), input ("Enter USN : ")

student. getMarks ()

student. display ()

Output:

Enter Student name : Iman

Enter USN : 4NCR3EE107

Enter marks in Subject 1 : 100

Enter marks in Subject 2 : 100

Enter marks in Subject 3 : 100

Student Name : Iman

USN : 4NCR3EE107

Marks : [100, 100, 100]

Total : 300

Percentage : 100.00

Grade : A



1R) Design GUI a python program to store the information about the student like first name, lastname, department (text box), gender (radio button) with login and cancel button.

Input :

```
from tkinter import *
```

```
def login_action():
```

```
    print(f"First Name: {first_name_entry.get()}")
```

```
    print(f"Last Name: {last_name_entry.get()}")
```

```
    print(f"Department: {department_entry.get()}")
```

```
    print(f"Gender: {'Male' if gender_var.get() == 1 else 'Female'}")
```

```
    print(f"Terms Accepted: {'Yes' if terms_var.get() else 'No'}")
```

```
def cancel_action():
```

```
    first_name_entry.delete(0, END)
```

```
    last_name_entry.delete(0, END)
```

```
    department_entry.delete(0, END)
```

```
    gender_var.set(0)
```

```
    terms_var.set(0)
```

```
root = Tk()
```

```
root.title("Student Information")
```

```
first_name_label = Label(root, text="First Name: ")
```

```
last_name_label = Label(root, text="Last Name: ")
```

```
department_label = Label(root, text="Department: ")
```

```
gender_label = Label(root, text="Gender: ")
```

```
first_name_entry = Entry(root)
```

```
last_name_entry = Entry(root)
```

```
department_entry = Entry(root)
```

```
gender_var = IntVar()
```

```
male_radio = Radiobutton(root, text="Male", variable=gender_var, value=1)
```

female_radio = Radiobutton (root, text = "female", variable = gender_var, value = 2)

terms_var = IntVar()

terms_check = Checkbutton (root, text = "I agree to the terms and conditions", variable = terms_var)

login_button = Button (root, text = "Login", command = login_action)

cancel_button = Button (root, text = "Cancel", command = cancel_action)

first_name_label.grid (row = 0, column = 0)

first_name_entry.grid (row = 0, column = 1)

last_name_label.grid (row = 1, column = 0)

last_name_entry.grid (row = 1, column = 1)

department_label.grid (row = 2, column = 0)

department_entry.grid (row = 2, column = 1)

gender_label.grid (row = 3, column = 0)

male_radio.grid (row = 3, column = 1)

female_radio.grid (row = 3, column = 2)

terms_check.grid (row = 4, column = 0, columnspan = 2)

login_button.grid (row = 5, column = 0)

cancel_button.grid (row = 5, column = 1)

root.mainloop()

Output:

First Name: Syed

Last Name: Iman

Department: Mathematics

Gender: Male

Terms Accepted: Yes

[When you execute the program, it will open a window titled "Student Information" containing the following elements arranged in a grid layout:

In which we have to write our first name, last name, department, and we have to choose gender either male or female, There is a checkbox for the terms to accept, then there'll be login or cancel options].



Date :

Experiment No.

13) Design a simple GUI image viewer

Input:

from tkinter import *

root = Tk()

image = PhotoImage (file='filename.png')

label = Label (root, image = image)

label.pack()

root.mainloop()

Output:

[Window with Image Display]

When you run this code, a new window will open displaying the image file name.png.

- The window will adjust its size to fit the dimensions of the image.
- If the image is too large for the window, scroll bars will appear to allow you to navigate through the image.
- The window will stay open and responsive until you manually close it.]

1H) Define a function that takes two objects representing complex numbers and return new complex numbers with an addition of two complex numbers. Define a suitable class 'Complex' to represent the complex number. Develop a program to read N ($N > 2$) complex numbers and to compute the addition of N complex numbers.

Input:

class Complex :

```
def __init__(self, real, img):
    self.real = real
    self.img = img
```

def __add__(self, other) :

```
    return Complex(self.real + other.real, self.img + other.img)
```

def __str__(self) :

```
    return f'{self.real} + {self.img}i'
```

$n = \text{int}(\text{input}(\text{"Enter the number of complex numbers : "}))$

numbers = []

for i in range(n) :

real = float(input("Enter the real part : "))

img = float(input("Enter the imaginary part : "))

numbers.append(Complex(real, img))

total = Complex(0, 0)

for number in numbers :

total = total + number

print("The total is : ", total)

Output:

Enter the number of complex numbers : 3

Enter the real part : 2

Enter the imaginary part : 3

Enter the real part : 5

Enter the imaginary part : -3

Enter the real part : 5

Enter the imaginary part : 6

The total is : 12.0 + 6.0i