PYTHON PROGRAMMING LAB

Program 1

Write a python program to create a grade calculator.

Aim

To write a python program to create a grade calculator.

Algorithm

Step1: Start the process.

Step2: Open PyCharm Community Edition 2021.3.2.

Step3: Prompt the user to enter the name, regno and marks scored for the respective subjects.

Step4: Calculate total marks and average.

Step5: Use conditional statements to determine the grade.

Step6: Display the result.

Steo7: stop the process.

Coding

```
print("Enter Name ")
name = input()
print("Enter Register Number: ")
reg_number = input()
print("Enter Marks Obtained in 5 Subjects: ")
markOne = int(input())
markTwo = int(input())
markThree = int(input())
markFour = int(input())
markFive = int(input())
tot = markOne+markTwo+markThree+markFour+markFive
avg = tot/5
if avg>=91 and avg<=100:
    print("Your Grade is A1")
elif avg>=81 and avg<91:
    print("Your Grade is A2")
elif avg>=71 and avg<81:
    print("Your Grade is B1")
```

```
elif avg>=61 and avg<71:
    print("Your Grade is B2")
elif avg>=51 and avg<61:
    print("Your Grade is C1")
elif avg>=41 and avg<51:
    print("Your Grade is C2")
elif avg>=33 and avg<41:
    print("Your Grade is D")
elif avg>=21 and avg<33:
    print("Your Grade is E1")
elif avg>=0 and avg<21:
    print("Your Grade is E2")
else:
    print("Invalid Input!"
```

Output

```
C:\Users\st\PycharmProjects\pythonProject\venv\bin\python.exe
C:/Users/st/PycharmProjects/pythonProject/main.py
Enter Name
ARUN
Enter Register Number:
22UCIT003
Enter Marks Obtained in 5 Subjects:
90
89
89
89
90
Your Grade is A2


Process finished with exit code 0
```

Result

Thus the program for grade calculator has been executed and verified successfully.

Program 2

Write a python program to display the numbers of series.

Aim

To write a python program to display the numbers of series.

Algorithm

Step1: Start the process.

Step2: Open PyCharm Community Edition 2021.3.2.

Step3: Prompt the user to enter a number(n) to determine the series range.

Step4: Initialize the empty list.

Step5: Generate and print number series using for loop.

Step6: Calculate and print sum.

Step7: Display the result.

Step8: Stop the process.

Coding

```
print(" Printing Number Series" )
print("**********************")
n = int(input("Enter a number: "))
a = []
for i in range(1, n + 1):
    print(i, sep=" ", end=" ")
    if (i < n):
        print("+", sep=" ", end=" ")
    a.append(i)
print("\n The sum of given number series is : ", sum(a))
```

Output

```
C:\Users\st\PycharmProjects\pythonProject3\venv\bin\python.exe
C:/Users/st/PycharmProjects/pythonProject3/main.py
 Printing Number Series
**********************
Enter a number: 4
1 + 2 + 3 + 4
The sum of given number series is: 10
Process finished with exit code 0
```

Result

Thus the program for number series has been executed and verified successfully.

Program 3

Write a python program to implement binary search algorithm.

Aim

To write a python program to implement binary search algorithm.

Algorithm

Step1: Start the process.

Step2: Open PyCharm Community Edition 2021.3.2.

Step3: Obtain a sorted array (arr), a target element(target) and initialize start to 0 and end to the length of the array minus 1.

Step4: Calculate the middle index as mid=(Start +end)//2.

Step5: If arr[mid] is equal to the target, return mid as the index where the target is found.

Step6: If arr[mid] less than the target, update end to mid-1

Step8: If the loop terminates without finding the target, return -1 to indicate that the target is not present in the array

Step9: Display the result.

Step10: Stop the process.

Coding

```
def binary_search(arr,start,end,target):
    mid = (start+end)//2
    if start > end:
        return -1
    if arr[mid] < target:
        return binary_search(arr, mid+1, end, target)
    elif arr[mid] > target:
        return binary_search(arr, start, mid-1, target)
    elif arr[mid] == target:
        return mid
```

```
arr = [20,30,40,60,80,90]
print("Enter the number to search: ")
a = int(input())
ans = binary_search(arr,0,len(arr)-1 ,a)
if ans  == -1:
    print("Target Not Found")
else:
    print("Target found at index",ans)
```

Output

```
C:\Users\st\PycharmProjects\pythonProject4\venv\bin\python.exe
C:/Users/st/PycharmProjects/pythonProject4/main.py
Enter the number to search:
30
Target found at index 1
Process finished with exit code 0

C:\Users\st\PycharmProjects\pythonProject4\venv\bin\python.exe
C:/Users/st/PycharmProjects/pythonProject4/main.py
Enter the number to search:
 3
Target Not Found

Process finished with exit code 0
```

Result

Thus the program for implementing binary search algorithm has been executed and verified successfully.

Program 4

Write a python program to check if the given string is pangram or not.

Aim

To write a python program to check if the given string is pangram or not.

Algorithm

Step1: Start the process.

Step2: Open PyCharm Community Edition 2021.3.2.

Step3: Import the string module.

Step4: Create a function called ispangram that takes string as a parameter.

Step5: Create a string alphabets containing all lowercase letters from ‘a’ to ‘z’.

Step6: Convert the input string to the lowercase.

Step7: Iterate through each character, if char is not in lowercase input string and return False; if the loop completes return True.

Step8: Take user input for a string and store it.

Step9: Call the ispangram function with user provided string, if the function returns true print Yes, else print No.

Step10: Stop the process.

Coding

```
import string
def ispangram(str):
  alphabet = "abcdefghijklmnopqrstuvwxyz"
  for char in alphabet:
    if char not in str.lower():
      return False
  return True
string = input("Enter String: ")
if(ispangram(string) == True):
  print("The Given String is pangram")
else:
  print("The Given String is not pangram")
```

Output

```
C:\Users\st\PycharmProjects\pythonProject5\venv\bin\python.exe
C:/Users/st/PycharmProjects/pythonProject5/main.py
Enter String: Computer Technology and Information Technology
The Given String is not pangram

Process finished with exit code 0

C:\Users\st\PycharmProjects\pythonProject5\venv\bin\python.exe
C:/Users/st/PycharmProjects/pythonProject5/main.py
Enter String: The five boxing wizards jump quickly
The Given String is pangram

Process finished with exit code 0
```

Result

Thus the program for checking pangram has been executed and verified successfully.

Program 5

Write a python program to create a list and perform the following methods: 1) insert() 2) remove() 3) append() 4) len() 5) pop() 6) clear()

Aim

To write a python program to create a list and perform the following methods: 1) insert() 2) remove() 3) append() 4) len() 5) pop() 6) clear()

Algorithm

Step1: Start the process.

Step2: Open PyCharm Community Edition 2021.3.2.

Step3: Initialize a list named fruits with initial values.

Step4: Print the item in the list.

Step5: Use len(fruits) to calculate and print the length of the list.

Step6: Use fruits.append(item) to add the item to the end of the list.

Step7: Use fruits.remove(item) to remove the specified item from the list.

Step8: Use fruits.insert(0,item) to insert the item at the beginning of the list.

Step9: Use popped = fruits.pop(index) to remove and print the item at the specified list.

Step10: Use fruits.clear to remove all items from the list.

Step11: Display the result.

Step12: Stop the process.

Coding

```
print("\t\t\t\t\t PROGRAM USING LIST METHODS")
fruits=['apple', 'orange', 'banana', 'mango']
print("The items in the list fruits are", fruits)
length = len(fruits)
print ("The length of the list is:", length)
string=input("Enter the item to append into the list:")
fruits.append(string)
print("The list after appended is:", fruits)
string=input("Enter the item to remove from the list:")
fruits.remove(string)
print("The list after item removed is:", fruits)
string=input("Enter the item to insert into the list:")
fruits.insert(0,string)
print("The list after item insertion is:", fruits)
string=input("Enter the item to pop up from the list:")
popelements=fruits.pop(0)
print("The popup  element is:", popelements)
print("The list after Popup is:", fruits)
```

Output

PROGRAM USING LIST METHODS

The items in the list fruits are ['apple', 'orange', 'banana', 'mango']

The length of the list is: 4

Enter the item to append into the list: jackfruit

The list after appended is: ['apple', 'orange', 'banana', 'mango', 'jackfruit']

Enter the item to remove from the list: jackfruit

The list after item removed is: ['apple', 'orange', 'banana', 'mango']

Enter the item to insert into the list: jackfruit

The list after item insertion is: ['jackfruit', 'apple', 'orange', 'banana', 'mango']

Enter the item to pop up from the list: jackfruit

The popup element is: jackfruit

The list after Popup is: ['apple', 'orange', 'banana', 'mango']

Process finished with exit code 0

Result

Thus the program for list manipulation has been executed and verified successfully.

Program 6

Write a python program to create a list of tuples with the first element as the number and second element as the square of the number.

Aim

To write a python program to create a list of tuples with the first element as the number and second element as the square of the number.

Algorithm

Step1: Start the process.

Step2: Open PyCharm Community Edition 2021.3.2.

Step3: Print ‘list of tuples’ to indicate the purpose of the program.

Step4: Initialize an empty list to store the tuples.

Step5: prompt the user to enter a number (‘n’).

Step6: Iterate over the numbers from 1 to ‘n’.

i)For each number, create a tuple with the first element as the number and second element as the square of the number.

ii)Append this tuple to the list

Step7: Display the result.

Step8: Stop the process.

Coding

```
print("List of Tuples")
def square_of_tuples(n):
    return [(i, i*i)
            for i in range(1, n+1)]
n=int(input("Enter the Number:"))
result = square_of_tuples(n)
print(result)
```

Output

```
C:\Users\pk\PycharmProjects\pgm6\venv\bin\python.exe
C:/Users/pk/PycharmProjects/pgm6/main.py
List of Tuples
Enter the Number:6
[(1, 1), (2, 4), (3, 9), (4, 16), (5, 25), (6, 36)]

Process finished with exit code 0
```

Result

Thus the program for tuples with the first element as the number and second element as the square of the number has been executes and verified successfully.

Program 7

Write a python program to combine two lists into a dictionary.

Aim

To write a python program to combine two lists into a dictionary.

Algorithm

Step1: Start the process.

Step2: Open PyCharm Community Edition 2021.3.2.

Step 3: Define a function that takes two parameters (keys and value)

Step4: Use zip function(zip()) to combine the two lists into pairs of key-value tuples.

Step5: Convert the resulting tuples into a dictionary using the dict constructor.

Step6: Take the user input for the lists.

Step7: Call the function with the user-input list as argument and store the result

Step8: Display the result.

Step9: Stop the process.

Coding

```
def lists_to_dictionary(keys, values):
    return dict(zip(keys, values))
keys = input("Enter the First list:")
values = input("Enter the Second list:")
result = lists_to_dictionary(keys, values)
print("The Elements present in the Dictionary are:")
print(result)
```

Output

```
C:\Users\TEMP.CTIT.067\PycharmProjects\pgm7\venv\bin\python.exe
C:/Users/TEMP.CTIT.067/PycharmProjects/pgm7/main.py
Enter the First list:12345
Enter the Second list: ABCDE
The Elements present in the Dictionary are:
{'1': 'A', '2': 'B', '3': 'C', '4': 'D', '5': 'E'}

Process finished with exit code 0
```

Result

Thus the program to combine two lists into a dictionary has been executed and verified successfully.

Program 8

Write a python program to append text to a file.

Aim

To write a python program to append text to a file.

Algorithm

Step1: Start the process.

Step2: Open PyCharm Community Edition 2021.3.2.

Step 3: Define a function that takes a file name as parameter.

Step4: Open the file in append mode(‘a+’)

Step5: prompt the user to enter the name of the city.

Step6: Close the file after writing

Step7: Reopen the file in read mode(‘r’)

Step8: print the content of the file.

Step 9: prompt user to enter the file name.

Step10: call the append file function with filename.

Step11: stop the process.

Coding

```
def appendfile(fname):
    f=open(fname,"a+")
    city=input("Enter the name of the city:")
    f.write(city+'\n')
    f.close()
    f=open(fname,'r')
    print(f.read())
    f.close()
file=input("Enter the name of the file:")
appendfile(file)
```

Output

```
E:\pgm8\venv\bin\python.exe E:/pgm8/main.py
Enter the name of the file:citydts.txt
Enter the name of the city: karnataka
canada
mumbai
kerala
karnataka

Process finished with exit code 0
```

Result

Thus the program to append text to a file has been executed and verified successfully.

Program 9

Write a python program to simulate the banking operations using class.

Aim

To write a python program to simulate the banking operations using class.

Algorithm

Step1: Start the process.

Step2: Open PyCharm Community Edition 2021.3.2.

Step3: prompt user to enter Account_number, Account_holder name, and initial balance.

Step4: create a variable balance and set it to initial balance.

Step5: enter a loop to repeatedly display a menu of options to the user.

Step6: Deposit

i)prompt the user to enter an amount to be deposit

ii)add the amount to the initial balance

step7: withdraw

i)prompt the user to enter an amount

ii) check if amount is less than or equal or equal to balance if yes, subtract the amount from balance; If no, display insufficient funds message.

Step8: Display the result.

Step9: Stop the process

Coding

```
class BankAccount:
    def __init__(self, account_number, balance):
        self.account_number = account_number
        self.balance = balance


def deposit(self, amount):
        self.balance += amount
        print(f"Deposited {amount} rupees.")
```

```
    def withdraw(self, amount):
        if amount <= self.balance:
            self.balance -= amount
            print(f"Withdraw {amount} rupees. ")
        else:
            print("Insufficient funds")

    def display_balance(self):
        print(f"Account Number: {self.account_number}, Balance:
{self.balance} rupees")

# Getting input from the user
account_number = int(input("Enter account number: "))
account_name = input("Enter the account holder name: ")
initial_balance = float(input("Enter initial balance: "))
account = BankAccount(account_number, initial_balance)

while True:
    print("\n1. Deposit\n2. Withdraw\n3. Display Balance\n4. Exit")
    choice = input("Enter your choice (1/2/3/4): ")

    if choice == "1":
        amount = float(input("Enter the amount to deposit: "))
        account.deposit(amount)
    elif choice == "2":
        amount = float(input("Enter the amount to withdraw: "))
        account.withdraw(amount)
    elif choice == "3":
        account.display_balance()
```

```
    elif choice == "4":
        print("Exiting the program")
        break
    else:
        print("Invalid choice. Please try again.")
```

Output

```
C:\Users\st\PycharmProjects\pythonProject11\venv\bin\python.exe
C:/Users/st/PycharmProjects/pythonProject11/main.py
Enter account number: 123456
Enter the account holder name: RAJ
Enter initial balance: 4000

1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice (1/2/3/4): 1
Enter the amount to deposit: 3000
Deposited 3000.0 rupees.

1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice (1/2/3/4): 3
Account Number: 123456, Balance: 7000.0 rupees

1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice (1/2/3/4): 2
Enter the amount to withdraw: 2000
Withdraw 2000.0 rupees.
```

1. Deposit
2. Withdraw
3. Display Balance
4. Exit

Enter your choice (1/2/3/4): 3

Account Number: 123456, Balance: 5000.0 rupees

1. Deposit
2. Withdraw
3. Display Balance
4. Exit

Enter your choice (1/2/3/4): 4

Exiting the program

Process finished with exit code 0

Result

Thus the program to simulate the banking operations using class has been executed and verified successfully.