

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 user_name = input("Enter your name : ")
6 print(f"Welcome {user_name} To Python World.")
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 # Creation of list data structure
6 lst = [1,2,3,4,5,6,7]
7 print(lst)
8
9 # List data structure methods
10
11 lst.append(8)      # lst_name.append(value)
12 print("Append :",lst)
13
14 lst2 = [9,10]
15 print(lst2)
16 lst.extend(lst2)   # lst_name.extend(another_lst)
17 print("Extend :",lst)
18
19 lst.insert(0,0)    # lst_name.insert(index,value)
20 print("Insert :",lst)
21
22 lst.remove(10)    # lst_name.remove(value_to_remove)
23 print("Remove",lst)
24
25 lst2.clear()       # Accepts no argument but clears the entire list --
26 lst_name.clear()
27 print("Clear :",lst2)
28
29 print("Index: ",lst.index(5))          # lst_name.index(value_to_get_index)
30
31 print("Count :",lst.count(3))          # lst_name.count(value_to_count)
32
33 lst.sort()        # Arranges the entire list item in assending order -- lst_name.sort()
34 print("Sort :",lst)
35
36 lst.reverse()      # Arranges the entire list item in decending/reverse order --
37 lst_name.reverse()
38 print("Reverse :",lst)
39
40 lst3 = lst.copy()
41 print("Copy :",lst3)
42
43 print()
44 print()
45
46 # a. Demonstrate positive and negative indexing with python List.
47 large_lst = [1,23,133,3321,4,84]
48
49 print("Positive Indexing")
50 for i in range(len(large_lst)):
51     print(large_lst[i])
52
53 print("Negative Indexing")
54 count = len(large_lst) - 1
55 while count >= 0:
56     print(large_lst[count])
57     count-=1
58
59 # b. Demonstrate slicing operations on python List.
```

```
58 print("Slicing Operation")
59 print(large_lst[2:5])
60
61 # c. Demonstrate updation on List elements in python.
62 print("Updation on list element")
63 large_lst[0] = 43434
64 print(large_lst)
65
66 # d. Demonstrate deletion of a single python list element and multiple elements using
67 # slicing operator
67 print("Single element deletion from list")
68 large_lst.pop(3)
69 print(large_lst)
70
71 print("Multiple element deletion from list")
72 del(large_lst[3:5])
73 print(large_lst)
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 # creation of tuples
6 tup1 = (1,2,3,4,5,2,2)
7 print("This is a tuple : ",tup1)
8
9 # count() method of tuple
10 print("This is a tuple.count() program : ",tup1.count(2))
11
12 # index() method of tuple
13 print("This is a tuple.index() program : ",tup1.index(2))
14
15 # (a) +ve indexing of tuple
16 print("This is positive indexing of tuple :",tup1[5])
17
18 # (a) -ve indexing of tuple
19 print("This is negative indexing of tuple :",tup1[-5])
20
21 # (b) slicing of the tuple
22 print("This is a slicing of the tuple :",tup1[2:5])
23
24 # (c) updating of the tuple elements
25 tup1 = list(tup1)
26 print("The tuple is converted to list so that it can be updated.", tup1)
27 tup1[5] = 25
28 tup1 = tuple(tup1)
29 print("The list is updated and converted back to tuple :",tup1)
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 # Creation of the dictionary
6
7 thisdict = {
8     "brand": "Ford",
9     "model": "Mustang",
10    "year": 1964
11 }
12 print("This is a dictionary :",thisdict)
13
14 # (a) Updation of the dictionary
15 thisdict2 = {
16     "batch no.": "A42DFS"
17 }
18 thisdict.update(thisdict2)
19 print("This is the updation of the dictionary :",thisdict)
20
21 # (b) Removal of the elements of the dictionary
22 thisdict.pop("model")
23 print("The modal element is removed from the dictionary :",thisdict)
24
25 # (a) clear(), copy(), get(), items(), keys(), popitem(), and values().
26 thisdict2.clear()
27 print("The dictionary 2 is now cleared :",thisdict2)
28 thisdict3 = thisdict.copy()
29 print("The dict3 is a copy of the dict 1:",thisdict3)
30 print("get() function :",thisdict.get("year"))
31 print("items() function :",thisdict.items())
32 print("keys() function :",thisdict.keys())
33 thisdict.popitem()
34 print("popitem() function :",thisdict)
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 text_string = "MY NAME IS Saurabh Yadav. "
6
7 # lower()
8 print("lower() :",text_string.lower())
9
10 # Upper()
11 print("upper() :",text_string.upper())
12
13 # join()
14 lst = ["This","is","a","join()","function"]
15 print("join() :"," ".join(lst))
16
17 # Upper()
18 print("split() :",text_string.split())
19
20 # find()
21 text_string2 = "".join(text_string)
22 print("find() :",text_string2.find("Aman"))
23
24 # replace()
25 new_text = "I like bananas"
26 value = new_text.replace("bananas", "apples")
27 print(value)
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 print("Calculator1")
6 star = 148
7 print("*"*star)
8 print('''
9     1)Div
10    2)Mul
11    3)Add
12    4)Sub
13 ''')
14 print("*"*star)
15 while (True):
16     try:
17         operator = int(input("Enter operator : "))
18         break
19     except ValueError and NameError:
20         print("You have entered wrong input for operator\n")
21 while (True):
22     try:
23         num1 = float(input("Enter number 1 : "))
24         break
25     except ValueError and NameError:
26         print("You have entered wrong input for number 1\n")
27 while (True):
28     try:
29         num2 = float(input("Enter number 2 : "))
30         break
31     except ValueError and NameError:
32         print("You have entered wrong input for number 1\n")
33
34 if operator == 1:
35     ans = num1 / num2
36     print("The div is : ",ans)
37 elif operator == 2:
38     ans = num1 * num2
39     print("The mul is : ",ans)
40 elif operator == 3:
41     ans = num1 + num2
42     print("The sum is : ",ans)
43 elif operator == 4:
44     ans = num1 - num2
45     print("The sub is : ",ans)
46 else:
47     print("Invalid Operator")
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 def input_list(ele_num):
6     lst = []
7     for element in range(ele_num):
8         while True:
9             try:
10                 try:
11                     lst.append(int(input("Enter :")))
12                     break
13                 except Exception:
14                     print("Enter only number.")
15                 except KeyboardInterrupt:
16                     print("Enter the value to complete the list.\n")
17     return lst
18
19
20 def main():
21     ele_num = int(input("Enter the number of elements in list : "))
22     lst = input_list(ele_num)
23     while True:
24         try:
25             try:
26                 searchable = int(input("Enter the number you want to count : "))
27                 break
28             except Exception:
29                 print("Enter only number")
30         except:
31             print("Enter value first.")
32
33     print(f"The number {searchable} has occurred {lst.count(searchable)} times in the
34 entered list.")
35 main()
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 # initializing the content
6 lst = []
7
8 # min_max function for list items
9
10
11 def min_max(lst):
12     lst.sort(reverse=True)
13     return lst[len(lst)-1], lst[0]
14
15 def min_max_2(lst):
16     min = lst[0]
17     max = lst[1]
18     for index in range(len(lst)):
19         if min > lst[index]:
20             min = lst[index]
21         elif max < lst[index]:
22             max = lst[index]
23     return min,max
24
25 # Taking input from the user
26 for lst_index in range(int(input("Enter the no. of elements you want in the list : "))) :
27     while True:
28         try:
29             try:
30                 value = int(input("Enter the number : "))
31                 break
32             except KeyboardInterrupt:
33                 print("\nExiting.....\n")
34                 exit()
35             except Exception as e:
36                 print("\nEnter Only numbers.\n")
37         lst.append(value)
38
39 # taking min and max value of the list from min_max function
40 min, max = min_max(lst)
41
42 # Printing the min and max number of the list
43 print("Method 1")
44 print(f"The minimum number in the list is {min}")
45 print(f"The maximum number in the list is {max}")
46
47
48 # taking min and max value of the list from min_max function
49 min2, max2 = min_max_2(lst)
50
51 # Printing the min and max number of the list
52 print("\nMethod 2")
53 print(f"The minimum number in the list is {min2}")
54 print(f"The maximum number in the list is {max2}")
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 def row_column():
6     while True:
7         try:
8             try:
9                 row = int(input("Enter the number of rows of the matrix : "))
10                column = int(input("Enter the number of colunms of the matrix : "))
11                if row >= 1 and column >= 1:
12                    break
13                else:
14                    print("Enter number more than or equal to 1 for rows/column")
15            except Exception as e:
16                print("Enter only integer value for row/column")
17            except KeyboardInterrupt:
18                print("Invalid Key Binding")
19        return row, column
20
21 def choice_fun():
22     print('''The available functions are :\n
23         1) Addition
24         2) Subtraction
25         3) Multiplication
26         4) Transpose
27         5) Exit''')
28     while True:
29         try:
30             try:
31                 choice = int(input("Choice : "))
32                 if choice <= 5 and choice > 0:
33                     break
34                 else:
35                     print("Enter from above choice only")
36             except Exception as e:
37                 print("Invalid Input")
38             except KeyboardInterrupt:
39                 print("Invalid Key Binding")
40         return choice
41
42 def matrix_addition(matrix1,matrix2):
43     sum_matrix = []
44     for row_val in range(len(matrix1)):
45         lst = []
46         for col_val in range(len(matrix1[0])):
47             lst.append(matrix1[row_val][col_val] + matrix2[row_val][col_val])
48         sum_matrix.append(lst)
49     return sum_matrix
50
51 def matrix_subtraction(matrix1,matrix2):
52     sum_matrix = []
53     for row_val in range(len(matrix1)):
54         lst = []
55         for col_val in range(len(matrix1[0])):
56             lst.append(matrix1[row_val][col_val] - matrix2[row_val][col_val])
57         sum_matrix.append(lst)
58     return sum_matrix
59
```

```
60 def matrix_multiplication(matrix1,matrix2):
61     if len(matrix1[0]) == len(matrix2):
62         product_matrix = []
63         for col in range(len(matrix1[0])):
64             lst = []
65             for row in range(len(matrix1)):
66                 lst.append(0)
67             product_matrix.append(lst)
68         for i in range(len(matrix1)):
69             for j in range(len(matrix2[0])):
70                 for k in range(len(matrix2)):
71                     product_matrix[i][j] += matrix1[i][k] * matrix2[k][j]
72     return product_matrix
73 else:
74     print("The Multiplication is not possible\nas the number of columns of matrix
1\nis not equal to number of rows or matrix 2.")
75 main()
76
77 def matrix_transpose(matrix):
78     transpose_matrix = []
79     for i in range(len(matrix[0])):
80         lst = []
81         for j in range(len(matrix)):
82             lst.append(matrix[j][i])
83         transpose_matrix.append(lst)
84     return transpose_matrix
85
86 def input_matrix(row,column):
87     lst2 = []
88     for row_val in range(row):
89         lst = []
90         for col_val in range(column):
91             while True:
92                 try:
93                     try:
94                         lst.append(int(input("Enter : ")))
95                         break
96                     except Exception as e:
97                         print("\nEnter only integer value \n")
98                     except KeyboardInterrupt:
99                         print("Invalid Key Binding")
100                lst2.append(lst)
101    return lst2
102
103 def main():
104     choice = choice_fun()
105
106     if choice == 1:
107         print("Enter the values of the matrix first {0} will be the element of the
rows")
108         print("Matrix 1")
109         row_col = row_column()
110         matrix1 = input_matrix(row_col[0],row_col[1])
111         print("Matrix 2")
112         matrix2 = input_matrix(row_col[0],row_col[1])
113         sum_matrix = matrix_addition(matrix1,matrix2)
114         for row in range(len(sum_matrix)):
115             print(sum_matrix[row])
116         main()
117
```

```
118 elif choice == 2:  
119     print("Enter the values of the matrix first {0} will be the element of the  
rows")  
120     print("Matrix 1")  
121     row_col = row_column()  
122     matrix1 = input_matrix(row_col[0],row_col[1])  
123     print("Matrix 2")  
124     matrix2 = input_matrix(row_col[0],row_col[1])  
125     diff_matrix = matrix_subtraction(matrix1,matrix2)  
126     for row in range(len(diff_matrix)):  
127         print(diff_matrix[row])  
128     main()  
129  
130 elif choice == 3:  
131     print("Enter the values of the matrix first {0} will be the element of the  
rows")  
132     print("Matrix 1")  
133     row_col = row_column()  
134     matrix1 = input_matrix(row_col[0],row_col[1])  
135     print("Matrix 2")  
136     row_col = row_column()  
137     matrix2 = input_matrix(row_col[0],row_col[1])  
138     product_matrix = matrix_multiplication(matrix1,matrix2)  
139     for row in range(len(product_matrix)):  
140         print(product_matrix[row])  
141     main()  
142  
143 elif choice == 4:  
144     print("Enter the values of the matrix first {0} will be the element of the  
rows")  
145     row_col = row_column()  
146     print("Matrix")  
147     matrix = input_matrix(row_col[0],row_col[1])  
148     print("Original matrix")  
149     for row in range(len(matrix)):  
150         print(matrix[row])  
151     transpose_matrix = matrix_transpose(matrix)  
152     print("Transpose of a matrix")  
153     for row in range(len(transpose_matrix)):  
154         print(transpose_matrix[row])  
155     main()  
156  
157 else:  
158     print("Thank You for using the program... \nSee you later\nExiting.....")  
159     exit()  
160  
161 main()
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 # basic function for arithmetic operations
6 def add(num1,num2):
7     return num1 + num2
8 def sub(num1,num2):
9     return num1 - num2
10 def mul(num1,num2):
11     return num1 * num2
12 def div(num1,num2):
13     return num1 / num2
14
15
16 # function to take input
17 def input_function():
18     while (True):
19         try:
20             num1 = float(input("Enter number 1 : "))
21             break
22         except ValueError and NameError:
23             print("You have entered wrong input for number 1\n")
24     while (True):
25         try:
26             num2 = float(input("Enter number 2 : "))
27             break
28         except ValueError and NameError:
29             print("You have entered wrong input for number 1\n")
30     return float(num1),float(num2)
31
32
33 # main loop of a program
34 def main():
35     star = 148
36     print("*"*star)
37     print('''
38         1) Div
39         2) Mul
40         3) Add
41         4) Sub
42         5) Exit
43     ''')
44     print("*"*star)
45     while (True):
46         try:
47             operator = int(input("Enter operator : "))
48             break
49         except ValueError and NameError:
50             print("You have entered wrong input for operator\n")
51     if operator == 1:
52         num1,num2 = input_function()
53         ans = div(num1,num2)
54         print("The div is : ",ans)
55         main()
56     elif operator == 2:
57         num1,num2 = input_function()
58         ans = mul(num1,num2)
59         print("The mul is : ",ans)
```

```
60     main()
61 elif operator == 3:
62     num1,num2 = input_function()
63     ans = add(num1,num2)
64     print("The sum is : ",ans)
65     main()
66 elif operator == 4:
67     num1,num2 = input_function()
68     ans = sub(num1,num2)
69     print("The sub is : ",ans)
70     main()
71 elif operator == 5:
72     print("Thanks for using.")
73     exit()
74
75 if __name__ == "__main__":
76     main()
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 # Arbitrary Function
6 def greet(*names):
7     """This function greets all
8     the person in the names tuple."""
9
10    # names is a tuple with arguments
11    for name in names:
12        print("Hello", name)
13
14
15 greet("Saurabh", "Rakesh", "Pradeep", "Manish")
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 class student:
6     def get_data(self,name,age,branch,city):
7         self.std_name = name
8         self.std_age = age
9         self.std_branch = branch
10        self.std_city = city
11
12    def display(self):
13        print("Student name :",self.std_name)
14        print("Stutent age :",self.std_age)
15        print("Student branch :",self.std_branch)
16        print("Student city :",self.std_city)
17
18 std_obj = student()
19 std_obj.get_data("saurabh",18,"B.Tech CSE","Vapi")
20 std_obj.display()
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 class BankAccount:
6
7     # Initial Function
8     def __init__(self,AccNumber,AccHolder,AccBalance):
9         self.AccNumber = AccNumber
10        self.AccHolder = AccHolder
11        self.AccBalance = AccBalance
12
13    # Deposit Function
14    def DepositBalance(self,DepositAmount):
15        self.AccBalance+=DepositAmount
16        print("*"*100)
17        print("The Amount Credited.\nThe Account Balance is : ",self.AccBalance)
18
19    # Withdrawal Function
20    def WithdrawBalance(self,WithdrawAmount):
21        if (WithdrawAmount > self.AccBalance):
22            print("*"*100)
23            print("You don't have enough balance.")
24            print("*"*100)
25        else:
26            self.AccBalance-=WithdrawAmount
27        print("*"*100)
28        print("The Amount Debited.\nThe Account Balance is : ",self.AccBalance)
29
30    # Displaying the details of the Account Holder
31    def Display(self):
32        print("*"*100)
33        print("Account Number : ",self.AccNumber)
34        print("Account Holder : ",self.AccHolder)
35        print("Account Balance : ",self.AccBalance)
36
37 # Taking details of the Account Holder (Function)
38 def PersonDetails():
39     while True:
40         try:
41             AccNum = int(input("Enter Account Number : "))
42             AccName = input("Enter Person Name : ")
43             AccBal = float(input("Enter Account Balance : "))
44             break
45         except Exception:
46             print("Invalid Details...\\n")
47     return AccNum,AccName,AccBal
48
49 # Storing the values of the Person in the below variables
50 AccNum1,AccName1,AccBal1 = PersonDetails()
51
52 # Person 1 Details
53 F_Person = BankAccount(AccNum1,AccName1,AccBal1)
54 # Person 1 Account Deposit
55 print("*"*100)
56 F_Person.DepositBalance(float(input("Enter Deposit Ammount : ")))
57 print("*"*100)
58 # Person 1 Account Withdraw
59 F_Person.WithdrawBalance(float(input("Enter Withdrawal Ammount : ")))
```

```
60  
61 # displaying the Details for person 1  
62 F_Person.Display()  
63 print("*"*100)
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 class Employee:
6     emp_name = None
7     emp_age = None
8     emp_city = None
9     def get_data(self,emp_name,emp_age,emp_city):
10         self.emp_name = emp_name
11         self.emp_age = emp_age
12         self.emp_city = emp_city
13
14 class emp_derived(Employee):
15     def __init__(self,obj):
16         print("The employee name is :",obj.emp_name)
17         print("The employee age is  :",obj.emp_age)
18         print("The employee lives in  :",obj.emp_city)
19
20 obj = Employee()
21 obj.get_data("Saurabh",18,"Vapi")
22 emp_obj = emp_derived(obj)
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 class university:
6     name = None
7     year_of_estd = None
8     city = None
9     def __init__(self, name, year_of_estd, city):
10         self.name = name
11         self.year_of_estd = year_of_estd
12         self.city = city
13
14 class professor(university):
15     def __init__(self,
16                  designation, highest_qualification, area_of_research, year_of_experience, name_of_institute):
17         self.designation = designation
18         self.highest_qualification = highest_qualification
19         self.area_of_research = area_of_research
20         self.year_of_experience = year_of_experience
21         self.name_of_institute = name_of_institute
22
23     def display(self, obj):
24         print("Designation : ", self.designation)
25         print("Name : ", obj.name)
26         print("Year of joining : ", obj.year_of_estd)
27         print("Highest Qualification : ", self.highest_qualification)
28         print("Area of research : ", self.area_of_research)
29         print("Year of experience : ", self.year_of_experience)
30         print("Name of Institute : ", self.name_of_institute)
31         print("City : ", obj.city)
32
33 class lab_assistant(university):
34     designation = "Lab Assistant"
35
36     def __init__(self, highest_qualification, additional_skills, year_of_joining,
37                  name_of_institute):
38         self.highest_qualification = highest_qualification
39         self.additional_skills = additional_skills
40         self.year_of_joining = year_of_joining
41         self.name_of_institute = name_of_institute
42
43     def display(self, obj):
44         print("Designation : ", self.designation)
45         print("Name : ", obj.name)
46         print("Year of establish : ", obj.year_of_estd)
47         print("Year of joining", self.year_of_joining)
48         print("Highest Qualification : ", self.highest_qualification)
49         print("Name of institute : ", self.name_of_institute)
50         print("Additional Skills : ", self.additional_skills)
51         print("City : ", obj.city)
52
53 class office_assistant(university):
54     designation = "Office_assistant"
55
56     def __init__(self, qualification, year_of_joining, name_of_institute):
57         self.qualification = qualification
58         self.year_of_joining = year_of_joining
```

```
57     self.name_of_instituate = name_of_institute
58
59 def display(self,obj):
60     print("Designation : ", self.designation)
61     print("Name : ", obj.name)
62     print("Qualification : ", self.qualification)
63     print("Year of establish : ", obj.year_of_estd)
64     print("Year of joining : ", self.year_of_joining)
65     print("Name of institute : ", self.name_of_instituate)
66     print(obj.city)
67
68 class peon(university):
69     designation = "Office Peon"
70
71     def __init__(self,qualification, year_of_joining, name_of_institute):
72         self.qualification = qualification
73         self.year_of_joining = year_of_joining
74         self.name_of_instituate = name_of_institute
75
76     def display(self,obj):
77         print("Designation : ", self.designation)
78         print("Name : ", obj.name)
79         print("Qualification : ", self.qualification)
80         print("Year of establish : ", obj.year_of_estd)
81         print("Year of joining : ", self.year_of_joining)
82         print("Name of institute : ", self.name_of_instituate)
83         print("City : ", obj.city)
84
85
86
87 def main():
88     print('''Choose the option for the value you want to enter :
89     1) Professor
90     2) lab Assistant
91     3) Office Assisntt
92     4) Peon
93     5) Exit''')
94     while True:
95         try:
96             try:
97                 option = int(input("Enter the option : "))
98                 break
99             except Exception:
100                 print("Invalid Input")
101             except KeyboardInterrupt:
102                 print("Invalid Input")
103
104         if option == 1:
105             while True:
106                 try:
107                     obj = university(input("Employee Name : "),int(input("Year of
establish : ")),input("Employee City : "))
108                     obj_2 = professor(input("Enter the Designation : "),input("Enter the
Highest Qualification : "),input("Enter area of research : "),int(input("Year of
experience : ")),input("Name of the Institute : "))
109                     break
110                 except Exception:
111                     print("Please Enter the correct Details for every Field\nPlease Choose
the option again")
112                     obj_2.display(obj)
```

```
113
114     elif option == 2:
115         while True:
116             try:
117                 obj = university(input("Employee Name : "),int(input("Year of
118 establish : ")),input("Employee City : "))
119                 obj_2 = lab_assistant(input("Enter the Highest Qualification :
120 "),input("Enter additional skills : "),int(input("Year of joining : ")),input("Name
121 of the Institute : "))
122                 break
123             except Exception:
124                 print("Please Enter the correct Details for every Field\nPlease Choose
125 the option again")
126                 obj_2.display(obj)
127             elif option == 3:
128                 while True:
129                     try:
130                         obj = university(input("Employee Name : "),int(input("Year of
131 establish : ")),input("Employee City : "))
132                         obj_2 = office_assistant(input("Enter the Highest Qualification :
133 "),int(input("Year of joining : ")),input("Name of the Institute : "))
134                         break
135                     except Exception:
136                         print("Please Enter the correct Details for every Field\nPlease Choose
137 the option again")
138                         obj_2.display(obj)
139                     elif option == 4:
140                         while True:
141                             try:
142                                 obj = university(input("Employee Name : "),int(input("Year of
143 establish : ")),input("Employee City : "))
144                                 obj_2 = office_assistant(input("Enter the Highest Qualification :
145 ")),int(input("Year of joining : ")),input("Name of the Institute : "))
146                                 break
147                             except Exception:
148                                 print("Please Enter the correct Details for every Field\nPlease Choose
149 the option again")
150                                 obj_2.display(obj)
151             elif option == 5:
152                 print("Thanks for using the program....")
153                 print("Exiting.....")
154                 exit()
155             else:
156                 print("Wrong Input Option")
157                 main()
158 if __name__ == "__main__":
159     main()
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 class c:
6     def __init__(self,learnings,professor):
7         self.learning_c = learnings
8         self.professor_c = professor
9 class python:
10    def __init__(self,learnings,professor):
11        self.learning_p = learnings
12        self.professor_p = professor
13 class web_designing:
14    def __init__(self,learnings,professor):
15        self.learning_w = learnings
16        self.professor_w = professor
17
18 class student(c,python,web_designing):
19     def __init__(self,enroll,name,course):
20         self.enroll = enroll
21         self.name = name
22         self.course = course
23         c.__init__(self,learnings_c,professor_c)
24         python.__init__(self,learnings_python,professor_python)
25         web_designing.__init__(self,learnings_web,professor_web)
26
27     def display(self):
28         print("")
29         print("Enrollment number of student : ",self.enroll)
30         print("Name of the student : ",self.name)
31         print("Course : ",self.course)
32         print("Learnings : ")
33         print("C : ",self.learning_c)
34         print("Python : ",self.learning_p)
35         print("Web Designing : ",self.learning_w)
36         print("Professors : ")
37         print("C : ",self.professor_c)
38         print("Python : ",self.professor_p)
39         print("Web Designing : ",self.professor_w)
40
41     def get_learnings(NumOfLearnings):
42         learnings = []
43         for amount in range(NumOfLearnings):
44             learnings.append(input(f"Enter Learning {amount + 1} : "))
45         return learnings
46     print("*"*100)
47 StudEnroll = input("Enter student enroll : ")
48 StudName = input("Enter student name : ")
49 StudCourse = input("Enter student course : ")
50
51 NumOfLearningsC = int(input("Enter number of learnings in C : "))
52 learnings_c = get_learnings(NumOfLearningsC)
53 professor_c = input("Enter professor name = ")
54
55 NumOfLearningsPython = int(input("Enter number of learnings in Python : "))
56 learnings_python = get_learnings(NumOfLearningsPython)
57 professor_python = input("Enter professor name = ")
58
59 NumOfLearningsWeb = int(input("Enter number of learnings in Web_Designing : "))
```

```
60 learnings_web = get_learnings(NumOfLearningsWeb)
61 professor_web = input("Enter professor name = ")
62
63
64 data1 = c(learnings_c,professor_c)
65 data2 = python(learnings_python,professor_python)
66 data3 = web_designing(learnings_web,professor_web)
67
68 student_obj = student(StudEnroll,StudName,StudName)
69 student_obj.display()
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 class Base():
6     # Below is hidden classVariable
7     __ThisIsHidden = "I am a Hidden Varaible"
8     ThisIsNotHidden = "I am not a Hidden Varaible"
9
10 obj = Base()
11 print(obj.ThisIsNotHidden)
12
13 # The below statement will give an AttributeError as __ThisIsHidden is hidden inside
14 # the Base() Class
14 print(obj.__ThisIsHidden())
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 class area:
6     def find_area(self, *args):
7         if len(args) == 0:
8             return "Invalid Input"
9         elif len(args) == 2:
10            if args[0] == "square":
11                return (args[1] ** 2)
12            elif args[0] == "circle":
13                PI = 3.141
14                return (PI * (args[1] ** 2))
15            elif len(args) == 3:
16                if args[0] == "triangle":
17                    return ((args[1] * args[2]) / 2)
18                elif args[0] == "rectangle":
19                    return (args[1] * args[2])
20
21
22 shape_area = area()
23 square_area = shape_area.find_area("square", 5)
24 circle_area = shape_area.find_area("circle", 10)
25 triangle_area = shape_area.find_area("triangle", 5, 2)
26 rectangle_area = shape_area.find_area("rectangle", 5, 10)
27 print("Area of square :", square_area)
28 print("Area of circle :", circle_area)
29 print("Area of triangle :", triangle_area)
30 print("Area of rectangle :", rectangle_area)
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 class Parent():
6
7     def __init__(self):
8         self.value = "Inside from Parent"
9
10    def show(self):
11        print(self.value)
12
13 class Child(Parent):
14
15     # As soon as a function or attribute is defined in the child it is overridden by it
16     def __init__(self):
17         self.value = "Inside from Child"
18
19     def show(self):
20         print(self.value)
21
22
23
24 obj1 = Parent()
25 obj2 = Child()
26
27 obj1.show()
28 obj2.show()
29
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 try:
6     int_val = int(input("Enter any : "))
7     print("The Value is an integer value")
8 except Exception:
9     print("The value is not an integer")
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 while True:
6     try:
7         num = int(input("Enter the value : "))
8         break
9     except KeyboardInterrupt:
10        print("Enter the value first and then enter.")
11 if num % 2 != 0:
12    # if the condition above is true the below exception is raised
13    raise Exception("The number shouldn't be an odd integer")
14 else:
15    print("The program executed properly.")
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 def divide(x, y):
6     try:
7         # Floor Division : Gives only Fractional
8         # Part as Answer
9         result = x // y
10    except ZeroDivisionError:
11        print("Sorry ! You are dividing by zero ")
12    else:
13        print("Yeah ! Your answer is :", result)
14    finally:
15        # this block is always executed
16        # regardless of exception generation.
17        print('This is always executed')
18
19 # Look at parameters and note the working of Program
20 divide(3, 2)
21 divide(3, 0)
22
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 def global_finder(text):
6     def make_list(file_name):
7         # This reads the lines of the text file
8         try:
9             with open(file_name, "r") as file:
10                 lst = file.readlines()
11         except Exception:
12             print(f"Unable to locate file {file_name}.")
13             main()
14
15         # This removes spaces between the elements and appends it into lst2
16         lst2 = []
17         for i in lst:
18             lst2.append(i.split())
19
20         # This takes the individual elements from lst2 and extend the lst3
21         lst3 = []
22         for fin_ele in lst2:
23             lst3.extend(fin_ele)
24
25         # This removes any extra character like .,\\"() from any element in the
list3
26         lst4 = []
27         for clear_txt_ele in lst3:
28             lst4.append(clear_txt_ele.strip(".,\\" "()"))
29
30         # This converts the list4 elements into lower case characters
31         lst5 = []
32         for lower_ele in lst4:
33             lst5.append(lower_ele.lower())
34
35         # This returns the list5
36         return lst5
37
38     # This simply scans the list and returns common string with the other list
containing
39     # the number of times the element has occurred in the list.
40     def counter(wiki_lst):
41         lst = []
42         count_lst = []
43         for index_1 in range(len(wiki_lst)):
44             for index_2 in range(len(wiki_lst)):
45                 if wiki_lst[index_1] == wiki_lst[index_2]:
46                     if wiki_lst[index_1] not in lst:
47                         lst.append(wiki_lst[index_1])
48
49             for num_count in range(len(lst)):
50                 count_lst.append(wiki_lst.count(lst[num_count]))
51         return lst, count_lst
52
53     # Main function to control the flow of the program
54     def check():
55         lst = make_list(text)
56         element, value = counter(lst)
57         for element_index in range(len(element)):
```

```
58     print(element[element_index], ":", value[element_index])
59
60     while True:
61         try:
62             choice = input("Do you want to continue (Y/N): ")
63             break
64         except Exception:
65             print("\nEnter a valid input and then enter")
66
67     if choice == 'N' or choice == "n":
68         print("Exiting the program.")
69         exit()
70     else:
71         main()
72
73 check()
74
75 # Main function to control the flow of the program
76 def main():
77     print("\nPlease enter file extension")
78     while True:
79         try:
80             try:
81                 file_name = input("Enter the file name : ")
82                 if file_name.endswith(".txt"):
83                     break
84                 else:
85                     print("Enter file name that ends with .txt")
86             except Exception:
87                 print("Please enter valid input.")
88             except KeyboardInterrupt:
89                 print("Please enter a value first and then enter.")
90         try:
91             global_finder(file_name)
92         except Exception:
93             print("Unable to locate file.")
94
95 if __name__ == "__main__":
96     main()
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 print("Please enter file extension")
6 while True:
7     try:
8         try:
9             file_name = input("Enter the file name : ")
10            if file_name.endswith(".txt"):
11                break
12            else:
13                print("Enter file name that ends with .txt")
14        except Exception:
15            print("Please enter valid input.")
16    except KeyboardInterrupt:
17        print("Please enter a value first and then enter.")
18
19
20 while True:
21     try:
22         try:
23             reading_lines = int(input("Enter the number of lines you want to see :"))
24             if reading_lines > 0:
25                 break
26             else:
27                 print("Enter only natural numbers")
28         except Exception:
29             print("Please enter valid input.")
30     except KeyboardInterrupt:
31         print("Please enter a value first and then enter.")
32
33
34 with open(file_name, "r") as file:
35     file_lines_list = file.readlines()
36
37     print("\nThe particular lines are : \n")
38     line = 0
39     while reading_lines > line:
40         print(f"{line+1} {file_lines_list[line]}")
41         line+=1
42     file.close()
43
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 while True:
6     try:
7         try:
8             count = int(input("Enter the number of courses you want to add : "))
9             if count > 0:
10                 break
11             else:
12                 print("Enter only natural numbers")
13         except Exception:
14             print("Please enter valid input.")
15     except KeyboardInterrupt:
16         print("Please enter a value first and then enter.")
17
18 index_count = 0
19 course_list = []
20 while count > index_count:
21     course_list.append(input(f"Enter course {index_count+1} name : "))
22     index_count+=1
23
24 with open("program25.txt","w") as course_file:
25     for index in range(len(course_list)):
26         course_file.write(f"Course {index+1} : {course_list[index]}\n")
27 course_file.close()
28
29 print("The course names are saved into program25.txt file")
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 def get_files():
6     lst = ["source", "destination"]
7     file_name_lst = []
8     print("Please enter file extension")
9     for file_name_number in range(2):
10         while True:
11             try:
12                 file_name = input(f"Enter the {lst[file_name_number]} file name : ")
13                 if file_name.endswith(".txt"):
14                     file_name_lst.append(file_name)
15                     break
16                 else:
17                     print("The file must be a text file")
18             except KeyboardInterrupt:
19                 print("Enter file name and then enter.")
20     return file_name_lst[0], file_name_lst[1]
21
22 # Taking Source file name and Destination file name
23 src_file_name, dec_file_name = get_files()
24
25 with open(src_file_name, "r") as src_file:
26     src_content = src_file.readlines()
27     with open(dec_file_name, "w") as dec_file:
28         for line in src_content:
29             dec_file.write(line)
30     src_file.close()
31     dec_file.close()
32
33 print(f"The content of the file {src_file_name} is copied to {dec_file_name}.")
```

```
1 # B-Tech Computer Science And Technology
2 # Enro:- 202103103510362
3 # Name:- Saurabh Yadav
4
5 import string as st
6
7 file_start_lst = list(st.ascii_uppercase)
8 for file_prefix in file_start_lst:
9     file_name = file_prefix + ".txt"
10    with open(f"outputfolder/{file_name}", "w") as file:
11        print(f"The file {file_name} created sucessfully.")
```