

# AMRITSAR GROUP OF COLLEGES

Autonomous status conferred by UGC under UGC act-1956, (2f), NAAC-A Grade

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

B. Tech. (CSE2) 4<sup>th</sup> SEM

PROGRAMMING IN PYTHON (ACCS-16404)

### ASSIGNMENT -2

Student Name Sumit Kumar Giri Class: CSE2

Semester: 4<sup>th</sup>

University Roll No. 2000213

Total Marks :24

#### Section-A (6 Ques\* 2 Marks=12)

Ques 1.

Ques a: Describe the purpose of super keyword \_\_init\_\_ in Python. [CO 4]

Answer: The Super fun Key function in Python makes class inheritance more manageable and extensible. The function returns a temporary object that allows references to a parent class. It also avoid the usages of the parent class explicitly and also enable the multiple inheritance

Ques b: WAP to find union, intersection, difference and symmetric difference on frozen sets.[CO 3]

The value of set1: 11,12,13,14,15

The value of set2: 11,12,33,44,55

Answer: set1 = {11,12,13,14,15}

set2 = {11,12,33,44,55}

print("set1 union of set2:", set1.union(set2))

print("set1 intersection of set2:", set1.intersection(set2))

print("set1 difference of set2:", set1.difference(set2))

print("set1 symmetric difference of set2:", set1.symmetric\_difference(set2))

OUTPUT: - set1 union of set2: {33,11,12,13,14,15,44,55}

set1 intersection of set2: {11,12}

set1 difference of set2: {13,14,15}

set1 symmetric difference of set2: {33,14,55,13,44,15}

Ques c: Create an anonymous function to print the multiple of 10. [CO 1]

Answer  $a = \lambda x: x * 10$  for  $x$  in range(1, 11)

for  $a$  in  $a:$

print( $a$ )

OUTPUT :-

10

20

30

40

50

60

70

80

90

Ques d: Write down the various features of dictionaries in Python. [CO 3]

- Python's dictionaries are kind of hash table type.
- more than one entry per key not allowed, when duplicate encountered during assignment, the last assignment wins.
- keys must be immutable which means you can be use strings, numbers or tuples as dictionary keys but ['key'] is not allowed

Ques e: Why matplotlib Pyplot is used in Python? [CO 6]

Answer: matplotlib is cross-platform, data visualization and graphical plotting library for Python and its numerical extension numpy. It offers a viable open source alternative to MATLAB. It also use to embed plots in GUI application.

Ques f: Given a tuple pair ((4,5),(4,2),(9,8),(10,6)), count the number of pair(a,b) such that a and b are even. [CO 3]

Answer:  $tup = [(4,5), (4,2), (9,8), (10,6)]$

c = 0

for i in range(len(tup)):

if tup[i][0] % 2 == 0 and tup[i][1] % 2 == 0:

c += 1

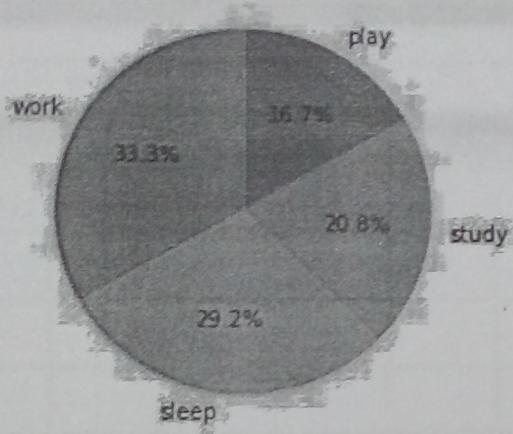
Print("The number of pair (a,b) such that a and b are even := ", c)

OUTPUT:- The number of pair (a,b) such that a and b are even = 2

Section-B (3 Ques\* 4 Marks=12)

Ques 2: Write a Python code to print the following output on screen. [CO 6]

Pie chart of daily activity.



Answer: let assume A is the list of contain the daily activity and list B show the time taken to complete the daily activity.

A = ['work', 'sleep', 'sleep', 'play']

B = [8, 7, 5, 4]

import matplotlib.pyplot as plt

A = ['work', 'sleep', 'study', 'play']

B = [8, 7, 5, 4]

plt.pie(B, labels=A, autopct='%.1f%%')

plt.title('pie chart of daily activity :')

pit.show()

Here, we have included the pyplot library as pit from the matplotlib.

import function. package by using the

Ques 3: Explain the various abstract data type available in Python. Write a Python code for the following operation: [CO 4]

14			14				
13	13		13	13			
12	12	12	12	12	12		
11	11	11	11	11	11	11	

Answer: The abstract datatype is special kind of data type whose behaviour is defined by a set of values and set of operation type. We can perform different operation but now these data operation are working that is totally hidden from the user. The API is made of primitive data type but operation logic are hidden some example

of ADI are stack, Queue list.

### STACK ADI

It is defined by the following structure and operation.  
A stack is structured as described above as an ordered collection of item where items are added and removed from the top stack are ordered LIFO.  
The stack operation are given :-  
(1) stack(): (2) push(item): (3) pop(): (4) peak():  
(5) IsEmpty(): (6) size():

### Queue Abstract datatype :-

A Queue is a fifo list with the following operation Enqueue, Dequeue, size, front.

- Queue():
- Enqueue (Item):
- dequeue():
- Front():
- IsEmpty():
- size():

of API are stack, Queue list.

List abstract datatype :-

The data is generally stored in key sequence in a line which has need structure consisting of Count, pointers and address of compare function needed to compare the data list.

- list():
- Insert():
- get():
- remove():
- remove all():
- replace():
- size():
- is Empty():
- is Full():

Ques 4: Explain inheritance in python with suitable example [CO 4]

Answer: Inheritance is the capability of one class to derive or inherit the properties from some other class. It is transitive in nature, which means that if class B inherits from another class A, then all the subclasses of B would automatically inherit from class A.

There are different form of inheritance:-

(i) Single inheritance :- when a child class inherits from only one parent class, it is called as single inheritance

Ex:- Class A:

```
def print(self):  
    print("Print1")
```

Class B(A):

```
def print2(self):  
    print("Print2")
```

b1 = B()

b1.print1()

b1.print2()

OUTPUT:-

Print1

Print2

(ii) multiple inheritance :- when a child class inherits from multiple parent classes, it is called as multiple inheritance.

Ex:- class Base1:

```
def b1(self):  
    print("Base1")
```

class Base2:

```
def b2(self):  
    print("Base2")
```

class Derived (Base1, Base2):

```
def hello(self):  
    print("Derived")
```

ob = Derived()

ob.hello()

ob.b1()

ob.b2()

OUTPUT:- Derived

Base1

Base2

(iii) Multilevel inheritance :- when we have child and grand child relationship.

Ex:- Class A:

```
def print1(self):  
    print("Point1")
```

Class B(A):

```
def print2(self):  
    print("Point2")
```

Class C(B):

```
def print3(self):  
    print("Point3")
```

```
ob = C()
```

```
ob.print1()
```

OUTPUT:-

```
Point1
```

(iv) Hierarchical inheritance :- Hierarchical inheritance more than one derived classes are created from a single base.

Ex:- Class Base1:

```
def print1(self):  
    print("Point1")
```

Class D1(Base1):

```
def print2(self):  
    print("Point2")
```

Class D2(Base1):

```
def print3(self):  
    print("Point3")
```

```
ob = D2()
```

```
ob.print1()
```

```
ob.print3()
```

OUTPUT:- Point1  
Point3

Hybrid inheritance :- This form combines more than one form of inheritance. Basically, it is a blend of more than type of inheritance.

Ex:- class A:

```
def print1(self):  
    print("Point1")
```

'S

-A Grade

```
class B(A):  
    def print2(self):  
        print("Point2")
```

'G

: 4th

```
class C(A):  
    def print3(self):  
        print("Point3")
```

'Mark

```
class D(B,A)  
    def print4(self):  
        print("Point4")
```

C

Ob = D()  
Ob. point1()  
Ob. point2()

OUTPUT:-      point1  
                  point2