Python Practicals Unit 1 and 2.

Q1. Write a python program to print date, time for today and now.

```
import datetime
a = print("Todays date is :",datetime.date.today()) b
= print("Now time is: ",datetime.datetime.now())
```

Output:-

```
Todays date is : 2024-02-05
Now time is:  2024-02-05 18:19:31.940967
```

2) Write a python program implement the arithmetic operator using module.

```
def add(x,y):
return x+y def
subs(x,y):
return x-y def
 product(x,y):
   return x*y
def div(x,y):
return x/y def
mod(x,y):
return x%y
```

```
add = print(add(20,40))
subs = print(subs(40,20))
prod = print(product(2,5))
div = print(div(20,5)) mod
= print(mod(20,5))
```

Output:-

```
60
20
10
4.0
 0
```

Q3. Write python program to create module calculating the area of a circle, Square and rectangle and call it into another program.

File1:

```
import math def
square(side):
return (side*side)
def circle(radius):
   return math.pi*(radius*radius) def
rectangle_area(length,breadth):
   return length * breadth
```

File2 :

```
import mod as dt


square_area = print("Area of square is: ",mod.square(5)) circle_area
= print("Area of cirlce is: ",mod.circle(5))
rect_area = print("Area of reactangle is: ",mod.rectangle_area(6,5))
```

Output:

```
Area of square is:  25
Area of cirlce is:  78.53981633974483
Area of reactangle is:  30
```

4)Write a Python program to generate a float between 10 and 20, inclusive and generate a random float within a specific range. Use random.uniform()

```
from random import *
print(randint(10, 20))  list1 =
[30, 23, 45, 16, 89, 56]
print(choice(list1))
print(uniform(10, 20))
```

Output:

```
18
23
12.787718773154166
```

Q5. Write a Python program generate square root and factorial of given number using module.

```
import math  square_root = print(“The square root of the
number is ”,sqrt(25)) factorial =  print(“The factorial of the
number is ”,factorial(5))
```

Output:

```
The square root of the number is 5.0
The factorial of the number is 120
```

Q6. Write a Python program to select a random element from a list, Use random.choice().

```
from random import *
list1 = [30, 23, 45, 16, 89, 56]
print(choice(list1))
```

Output:

```
16
```

Q7. Write a python program to implement any five dictionary function.

```
dict = {1:"raj", 2 : "Vishal", 3 : "Om"} print("The
length of dict is :", len(dict)) dict_copy =
dict.copy()
print("This is a copy of dict :", dict_copy)
print("The keys in the dict are: ",dict.keys())
print("The values in the dict are:", dict.values())
print("The items of the dict are :", dict.items())
```

Output:

```
The length of dict is : 3
This is a copy of dict : {1: 'raj', 2: 'Vishal', 3: 'Om'}
The keys in the dict are:  dict_keys([1, 2, 3])
The values in the dict are: dict_values(['raj', 'Vishal', 'Om'])
The items of the dict are : dict_items([(1, 'raj'), (2, 'Vishal'), (3, 'Om')])
```

Q8. Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are the square of the keys.

```
square_dict = {i: i**2 for i in range(1, 16)}
print("Dictionary with keys as numbers and values as squares:")
print(square_dict)
```

Output:

```
Dictionary with keys as numbers and values as squares:
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144,
13: 169, 14: 196, 15: 225}
```

Q9. Write a Python program to iterate over dictionaries using for loops.

```
my_dict = {'name': 'raj', 'age': 21, 'city': 'Jalgaon'}
print("Iterating over keys:") for key in my_dict:
   print(key)
print("\nIterating over values:")
for values in my_dict.values():
   print(values)
print("\nIterating over key-value pairs:")
for key, value in my_dict.items():
   print(f"{key}: {value}")
```

```
Output:  Iterating over values:
Prem
21
Jalgaon
Male

Iterating over keys:
name
age
city

Iterating over key-value pairs:
name: Bharatage:
21
city: Jalgaon
```

Q10. Write a Python class named Rectangle constructed from length and width and a method that will compute the area of a rectangle.

```
class rectangle:     def
area_rect(length,breadth):
       return length * breadth

area = print("The area of rectangle is : ",rectangle.area_rect(4,5))
```

Output:

```
The area of rectangle is :  20
```

Q11. Write a Python class Employee with attributes like emp_id, emp_name, emp_salary, and emp_department and print the information of employee.

```
class employee():     def
emp_info(emp_id,emp_name,emp_salary,emp_department):
print("The emp_id is :",emp_id)        print("The emp_name is
:",emp_name)         print("The emp_salary is :",emp_salary)
        print("The emp_department is :",emp_department)


emp_info = employee.emp_info(1,"Prem",70000,'IT SDE')
```

Output:

```
The emp_id is : 1
The emp_name is : Prem
The emp_salary is : 70000
The emp_department is : IT SDE
```

Q12. Write a python Program to demonstrate parameterized Constructor.

```
class Student:
def __init__(self,name):
        print("It's a parameterized
Constructor")
self.name = name
 def show(self):
        print("Hello ",self.name)



student = Student("Raj") student.show()
```

Output:

```
It's a parameterized Constructor
Hello  Raj
```

Q13. Write a python program to implement Inheritance- single, multiple, multilevel, hierarchical, hybrid.

Single:

```
class Vehicle:     def
Vehicle_info(self):
        print("Inside the parent
class") class Car(Vehicle):     def
car_info(self):
        print("Inside the derived class")

car= Car() car.Vehicle_info()
```

Output:

```
Inside the parent class
```

Multiple Inheritance:

```
class Person:     def person_info(self,
name, age):        print("Inside person
class")        print("Name : ",name," Age
:", age)

class Company:     def
company_info(self,company_name,location):
        print("Inside Company Class")        print("Name:
",company_name, ' location : ',location) class
Employee(Person, Company):     def Employe_info(self,
salary, skill):        print("Inside employee class")
print("Salary : ", salary , ' Skill : ',skill)

emp = Employee()

emp.person_info('Prem', 21)

emp.Employe_info('IBM', 'Pune')
```

Output:

```
Inside person class

Name :  Prem  Age : 21

Inside employee class

Salary :  IBM  Skill :  Pune
```

Multi-Level

```
class Animal:

def speak(self):

    print("Animal

Speaking")  class

Dog(Animal):      def

bark(self):

    print("dog barking")

class DogChild(Dog):

def eat(self):

    print("Eating bread...")
```

```
d = DogChild()
d.bark()
d.speak()
d.eat()
```

```
Output: dog
barking Animal
Speaking Eating
bread...
```

Hierarchical

```
class Vehicle:
def info(self):
    print("This is Vehicle") class
Car(Vehicle):    def car_info(self,
name):        print("Car name is:",
name) class Truck(Vehicle):    def
truck_info(self, name):
print("Truck name is:", name) obj1
= Car() obj1.info()
obj1.car_info('BMW') obj2 =
Truck() obj2.info()
obj2.truck_info('Ford')
```

Output:

```
This is Vehicle

Car name is: BMW

This is Vehicle

Truck name is: Ford
```

Hybrid

```
class Vehicle:     def
vehicle_info(self):
        print("Inside Vehicle
class") class Car(Vehicle):     def
car_info(self):
        print("Inside Car
class") class Truck(Vehicle):
def truck_info(self):
        print("Inside Truck class")
class SportsCar(Car, Vehicle):
def sports_car_info(self):
        print("Inside SportsCar class")
s_car = SportsCar()
s_car.vehicle_info() s_car.car_info()
s_car.sports_car_info()
```

Output:

```
Inside Vehicle class

Inside Car class

Inside SportsCar class
```

Q14. Write a program to demonstrate the working of Overloading Method.

```
Class Math:
   def add(self,a,b):
      print(a+b)

   def add(self,a,b,c):
      print(a+b+c)



math = Math()
math.add(5,6,7)
```

Output:

```
18
```

Q15. Python Program illustrate how to overload a binary + operator.

```
class Add:
    def __init__(self,a):
        self.a = a

    def __add__(self,o):
return self.a + o.a

ob1 = Add(1) ob2 = Add(2)
obj1 = Add("Raj") obj2 =
Add(" is  a Student")

print(ob1 + ob2) print(obj1
+ obj2)
```

Output:

```
3
Bharatis a Student
```