Exam Seat No:

Satish Pradhan Dnyanasadhana College

Thane

Certificate

This is to certify that Mr./Miss.: **Satish Vishwakarma** of **FYBSc Computer Science (Semester-II)** Class has successfully completed all the practical work in subject **Advance Python**, under the guidance of **Prof. Trupti Rongare** (subject in charge) during Year 2021-22 in partial fulfillment of Computer Science Practical Examination conducted by University of Mumbai.

Subject in charge **Head of the Department**

Date_____

Sr. No.	Index	Date	Sign
1	Write a program to Python program to implement various file operations.		
2	Write a Program to demonstrate concept of threading and multitasking in Python.		
3	Write a Python Program to work with databases in Python to perform operations such as a. Connecting to database b. Creating and dropping tables c. Inserting and updating into tables.		
4	Write a Python Program to demonstrate different types of exception handing.		
5	Write a GUI Program in Python to design application that demonstrates a. Different fonts and colours b. Different Layout Managers c. Event Handling		
6	Write Python Program to create application which uses date and time in Python.		
7	Write a program to Python program to implement concepts of OOP such as a. Types of Methods b. Inheritance c. Polymorphism		
8	Write a program to Python program to implement concepts of OOP such as a. Abstract methods and classes b. Interfaces		

1. Write a program to Python program to implement various file operations.

1.1) File reading

Code:

```
f = open("test.txt", "r")
print(f.read())
f.close()
```

Output:

```
Hello World, Its an test file
```

1.2) File Writing

Code:

```
f = open("testfile1.txt", "a")
f.write("Hello from the other side!!")
f.close()
f = open("testfile1.txt", "r")
print(f.read())
```

Output:

```
Hello from the other side!!
```

1.3) Creating and deleting file

Code:

```
import os
if os.path.exists("testfile1.txt"):
  os.remove("testfile1.txt")
  print("The file is removed")
else:
  print("The file does not exist creating new file")
  f = open("myfile.txt", "x")
  f.close()
```

Output:

```
The file is removed
```

2. Write a Program to demonstrate concept of threading and multitasking in Python.

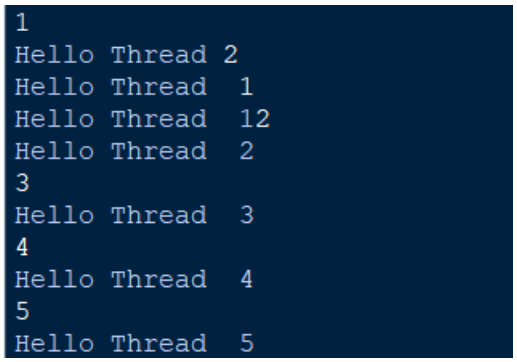
2.1) Threading

Code:

```
import _thread
def cT(tid):
   print("Hello Thread ",tid)
def pT():
   s = 0
   while True:
      s += 1
      _thread.start_new_thread(cT,(s,))
      if input()=="q":
         break
```

```
pT()
```

Output:



3. Write a Python Program to work with databases in Python to perform operations such as

a. **Connecting to database**
b. **Creating and dropping tables**
c. **Inserting and updating into tables.**

3.1) Connecting to database

Code:

```
import pymysql
db = pymysql.connect("localhost","root","12345","mydb")
cursor = db.cursor()
cursor.execute("SELECT VERSION()")
data = cursor.fetchone()
print("Database version: ", data)
db.close
```

3.2) Creating and dropping table

Code:

```
import pymysql
db = pymysql.connect("localhost","root","12345","mydb")
cursor = db.cursor()
cursor.execute("DROP TABLE IF EXISTS STUD")
cursor.execute("CREATE TABLE STUD(NAME VARCHAR(20),AGE INT, SEX
CHAR(1),INCOME FLOAT)")
db.close
```

3.3) Inserting data

Code:

```
import pymysql
db = pymysql.connect("localhost","root","12345","mydb")
cursor = db.cursor()
try:
    cursor.execute("INSERT INTO STUD VALUES ("RAM",18,"M",19868");)
    db.commit()
```

```
except:
        db.rollback()
db.close
```

3.4) Updating data

Code:

```
import pymysql
db = pymysql.connect("localhost","root","12345","mydb")
cursor = db.cursor()
try:
    cursor.execute("UPDATE STUD NAME="JARLISSON" WHERE INCOME=19868");)
    db.commit()
except:
        db.rollback()
db.close
```

4. Write a Python Program to demonstrate different types of exception handing.

Code:

```
try:
   a = int(input("Enter a:"))
   b = int(input("Enter b:"))
   c = a/b
   print("a/b = %d"%c)
except Exception:
   print("can't divide by zero")
   print(Exception)
else:
   print("Hi I am else block")
```

Output:

```
Enter a:13
Enter b:0
can't divide by zero
<class 'Exception'>
```

5. Write a GUI Program in Python to design application that demonstrates

a. **Different fonts and colours**
b. **Different Layout Managers**
c. **Event Handling**

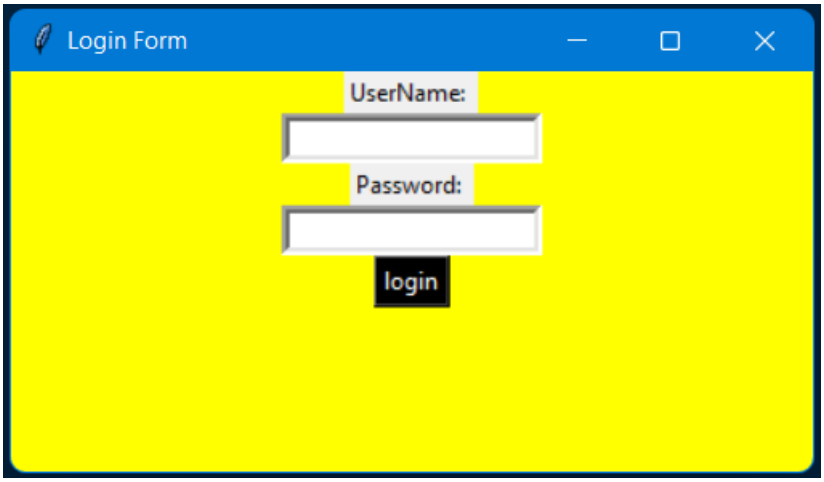
5.1) Login form

Code:

```
from tkinter import *
import tkinter as tk
from tkinter import messagebox
def loginCheck():
   if nameBox.get() == "VSatish":
```

```
        if PassBox.get() == "12345":
            messagebox.showinfo("Logim","Welcome!!")
        else:
            messagebox.showinfo("Logim","Invalid Password!")
win = tk.Tk()
win.geometry("400x200")
win.configure(bg="yellow")
win.title("Login Form")
name = Label(win, text="UserName: ")
name.pack()
nameBox = Entry(win,bd=4)
nameBox.pack()
Passw = Label(win, text="Password: ")
Passw.pack()
PassBox = Entry(win,bd=4,show="*")
PassBox.pack()
loginBtn = Button(win,text="login", command=loginCheck, bg="black",fg="white")
loginBtn.pack()
win.mainloop()
```

output:



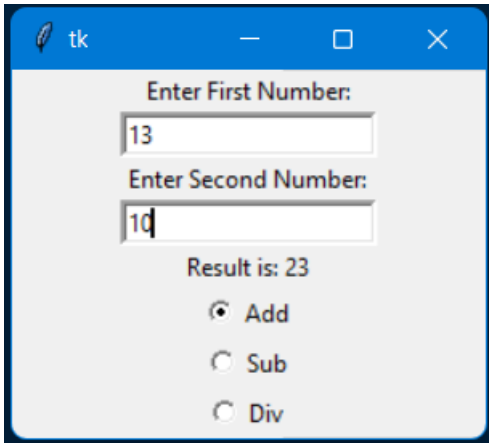
5.2) Simple Calculator

Code:

```
from tkinter import *
import tkinter as tk
def res():
    selection = var.get()
    t1 = int(txt1.get())
    t2 = int(txt2.get())
    if selection == 1:
        result = t1 + t2
    elif selection == 2:
        result = t1 - t2
    elif selection == 3:
```

```
        result = t1 / t2
    label.config(text="Result is: "+str(result))
win = tk.Tk()
var = IntVar()
lib1 = Label(win,text="Enter First Number:")
lib1.pack()
txt1 = Entry(win, bd=3)
txt1.pack()
lib2 = Label(win,text="Enter Second Number:")
lib2.pack()
txt2 = Entry(win, bd=3)
txt2.pack()
label = Label(win, text="Result")
label.pack()
r1 = Radiobutton(win,text="Add", variable = var,value = 1,command=res)
r1.pack()
r2 = Radiobutton(win,text="Sub", variable = var,value = 2,command=res)
r2.pack()
r3 = Radiobutton(win,text="Div", variable = var,value = 3,command=res)
r3.pack()
win.mainloop()
```

Output:

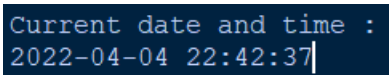


6. Write Python Program to create application which uses date and time in Python.

Code:

```
import datetime
now = datetime.datetime.now()
print ("Current date and time : ")
print (now.strftime("%Y-%m-%d %H:%M:%S"))
```

Output:



7. Write a program to Python program to implement concepts of OOP such as

a. **Types of Methods**
b. **Inheritance**
c. **Polymorphism**

7.1) Single inheritance

Code:

```
class student():
   def __init__(self, name, rollno):
      self.name = name
      self.roll = rollno
   def display(self):
      print(self.name)
      print(self.roll)
class detl(student):
   def __init__(self, name, roll, age, semn):
      self.age = age
      self.sem = semn
      student.__init__(self, name, roll)
a = detl("Sara", 90, 18, 2)
a.display()
```

Output:

```
Name:  Sara
Age:  90
```

7.2) Multiple inheritance

Code:

```
class Adhar():
   def __init__(self):
      self.AdharNo = 531565351
class Pan():
   def __init__(self):
      self.PanNo = "SJAD221S1AS12"
class Person(Adhar, Pan):
   def __init__(self,name):
      self.name = name
      Adhar.__init__(self)
      Pan.__init__(self)
   def getData(self):
      print("Name: ",self.name)
      print("Andhar No: ", self.AdharNo)
      print("Pan No: ", self.PanNo)
ob = Person("Gulzar")
ob.getData()
```

Output:

```
Name:  Gulzar
Andhar No:  531565351
Pan No:  SJAD221S1AS12
```

7.3) Multilevel inheritance

Code:

```
class Family:
    def show_family(self):
        print("Family:")
class Father(Family):
    fathername = ""
    def show_father(self):
        print(self.fathername)
class Mother(Family):
    mothername = ""
    def show_mother(self):
        print(self.mothername)
class Son(Father, Mother):
    def show_parent(self):
        print("Father :", self.fathername)
        print("Mother :", self.mothername)
s1 = Son()
s1.fathername = "Mark"
s1.mothername = "Sonia"
s1.show_family()
s1.show_parent()
```

Output:

```
Family:
Father : Mark
Mother : Sonia
```

7.4) Hierarchical inheritance

Code:

```
class Details:
    def __init__(self):
        self.__id=""
        self.__name=""
        self.__gender=""
    def setData(self,id,name,gender):
        self.__id=id
        self.__name=name
        self.__gender=gender
    def showData(self):
        print("Id: ",self.__id)
```

```
        print("Name: ", self.__name)
        print("Gender: ", self.__gender)
    class Employee(Details):
      def __init__(self):
        self.__company=""
        self.__dept=""
      def setEmployee(self,id,name,gender,comp,dept):
        self.setData(id,name,gender)
        self.__company=comp
        self.__dept=dept
      def showEmployee(self):
        self.showData()
        print("Company: ", self.__company)
        print("Department: ", self.__dept)
    class Doctor(Details):
      def __init__(self):
        self.__hospital=""
        self.__dept=""
      def setEmployee(self,id,name,gender,hos,dept):
        self.setData(id,name,gender)
        self.__hospital=hos
        self.__dept=dept
      def showEmployee(self):
        self.showData()
        print("Hospital: ", self.__hospital)
        print("Department: ", self.__dept)
    e=Employee()
    e.setEmployee(1,"Prem Sharma","Male","gmr","excavation")
    e.showEmployee()
    print("\n")
    d = Doctor()
    d.setEmployee(1, "pankaj", "male", "aiims", "eyes")
    d.showEmployee()
```

Output:

```
Id:  1
Name:  Prem Sharma
Gender:  Male
Company:  gmr
Department:  excavation


Id:  1
Name:  pankaj
Gender:  male
Hospital:  aiims
Department:  eyes
```

7.5) Hybrid inheritance

Code:

```
class University:
  def __init__(self):
    self.univ = "MU"
  def display(self):
    print("The University name is: ",self.univ)
class Course(University):
  def __init__(self):
    University.__init__(self)
    self.course = "CS"
  def display(self):
    print("The Course name is: ",self.course)
    University.display(self)
class Sem(University):
  def __init__(self):
    self.branch = 2
  def display(self):
    print("The Sem is: ",self.branch)
class Student(Course, Sem):
  def __init__(self):
    self.name = "Anshuman"
    Sem.__init__(self)
    Course.__init__(self)
  def display(self):
    print("The Name of the student is: ",self.name)
    Sem.display(self)
    Course.display(self)
ob = Student()
print()
ob.display()
```

Output:

```
The Name of the student is:  Anshuman
The Sem is:  2
The Course name is:  CS
The University name is:  MU
```

8. Write a program to Python program to implement concepts of OOP such as

a. Abstract methods and classes

b. Interfaces

Code:

```
from abc import ABC, abstractmethod
class Car(ABC):
   def mileage(self):
      pass
class Tesla(Car):
   def mileage(self):
      print("The mileage is 30kmph")
class Suzuki(Car):
   def mileage(self):
      print("The mileage is 25kmph ")
class Duster(Car):
    def mileage(self):
       print("The mileage is 24kmph ")
class Renault(Car):
   def mileage(self):
        print("The mileage is 27kmph ")
t= Tesla ()
t.mileage()
r = Renault()
r.mileage()
s = Suzuki()
s.mileage()
d = Duster()
d.mileage()
```

Output:

```
The mileage is 30kmph
The mileage is 27kmph
The mileage is 25kmph
The mileage is 24kmph
```