

# VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)  
(Affiliated to Osmania University)  
Hyderabad - 500 031.

DEPARTMENT OF

: CSE

NAME OF THE LABORATORY : PP LAB

Name K. Dree Indira Divani Roll No. 1602-21-733-052 Page No. 139

## PRELAB QUESTIONS: 18

1) What is inheritance? Give an example in python?

A: The technique of creating a new class from an existing class is called inheritance.

The old/existing class is known as base/parent class.

The new class is known as derived/child class.

→ Example:

```
• class person:  
    def __init__(self):  
        print("Base constructor")  
  
class student  
    def __init__(self):  
        super().__init__(self):  
        print("Derived constructor")
```

2) Differentiate between overloading and overriding.

Give an example in python.

A: Operator overloading allows programmers to extend the meaning of existing operators so that in addition to the basic data-types they can be also applied to user-defined data types.

# VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)  
(Affiliated to Osmania University)

Hyderabad - 500 031.

DEPARTMENT OF

: CSE

NAME OF THE LABORATORY : PP LAB

Name K.S.I.Sivani

Roll No. 1602-21-733-052 Page No. 140

A: \* Overriding: In python method overriding occurs by simply defining in the child class a method with the same name of method in the parent class.

3) How is data hiding achieved in python? Give an example:

A: Data hiding in python is performed using the -- double underscore before done prefix. This makes the class members non-public and isolated from the other classes.

Ex:

=  
class solution:

--privatecounter = 0

def sum(self):

self.--privatecounter += 1

print(self.--privatecounter).

count = Solution()

count.sum()

count.sum()

print(count.--privatecount)

4) What is \_\_repr\_\_() function in python? What is it used for? Give example:

# VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)  
(Affiliated to Osmania University)  
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : PPLAB

Name K.S.T.Sivani Roll No. 1602-21-733-052 Page No. 141

A Python \_\_repr\_\_() function returns the object representation in string format. This method is called repr() function is invoked on the object.

Ex: s = 'Hello'  
print(repr(s))  
print(repr(2.0/11.0))

5) Give examples for multiple inheritance and multilevel inheritance in python:

\* Multiple Inheritance

```
class Base1():
    def __init__(self):
        super(Base1, self).__init__()
        print("Base1 class")
```

```
class Base2():
    def __init__(self):
        super(Base2, self).__init__()
        print("Base2 class")
```

```
class Derived(Base1, Base2):
    def __init__(self):
        super(Derived, self).__init__()
        print("Derived class")
```

D = Derived()

# VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)  
(Affiliated to Osmania University)  
Hyderabad - 500 031.

DEPARTMENT OF

: CSE

NAME OF THE LABORATORY : PPLAB

Name K.S.P. Divani

Roll No. 1602-21-733052 Page No. 142

Multi-level Inheritance:

class Person:

```
def name(self):  
    print("Name...")
```

class Teacher(Person):

```
def qualification(self):  
    print("Qualification... Ph.D must")
```

class HOD(Teacher):

```
def experience(self):  
    print("Experience..... atleast 15 years")
```

hod = HOD()

hod.name()

hod.qualification()

hod.experience()

# VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)  
(Affiliated to Osmania University)  
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : PPLAB

Name K. Sree Indira Sivani Roll No. 1602-21-733-D52 Page No. 143

## PRELAB PROGRAMS: 12

- 1) Write a program that has classes such as student, course and department. Enroll a student in a course of a particular department.

class course:

```
def course_details(self):  
    self.course = input("Enter your name:")
```

class department:

```
def department_details(self):  
    self.department = input("Enter your department:")
```

class student(course, department):

```
def __init__(self):  
    self.name = input("Enter your name:")  
    self.course_details()  
    self.department_details()
```

stu = student()

O/P:

Enter your name: sivani  
Enter your course: python  
Enter your department: CSE

# VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)  
(Affiliated to Osmania University)

Hyderabad - 500 031.

DEPARTMENT OF

: CSE

NAME OF THE LABORATORY : PP LAB

Name K Sree India Savani Roll No. 1602-21-733-052 Page No. 144

- 2) Write a menu-driven program to overload +, - = and  
\*= operators on the matrix class.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : PPLAB

Name K.S.I. Sivani Roll No. 1602-21-733-D52 Page No. 145

- 3) Write a program with class Bill. The users have the option to pay the bill either by cheque or by cash. Use the inheritance to model this situation.

class Bill:

```
def check(self):  
    c = int(input("Enter your bill amount:"))
```

class cash(Bill):

```
def paycash(self):  
    print("You can pay the bill with cash")
```

class cheque(Bill):

```
def paycheque(self):  
    print("You can pay the bill with cheque")
```

print("Choose: 1. Cash 2. Cheque")

x = int(input("Choice:"))

if x == 1:

a = cash()

a.check()

a.paycash()

elif x == 2:

b = cheque()

b.check()

b.paycheque()

else:

pass

Output:

Choose: 1. Cash 2. Cheque

Choice: 2

Enter your bill amount: 1000

You can pay the bill with  
cheque.

# VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)  
(Affiliated to Osmania University)  
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : PP LAB

Name K.Sree Indra Sivani Roll No. 1602-21-733-052 Page No. 146

## \* LAB PROGRAMS - 12

- 1) Write a program that has a class Person. Inherit a class faculty from person. Create the instances of faculty.

class person:

```
def __init__(self, name, age):  
    self.name = name  
    self.age = age
```

```
def display(self):  
    print("Name:", self.name)  
    print("Age:", self.age)
```

class faculty(person):

```
def __init__(self, name, age, empid, experience):  
    super().__init__(name, age)  
    self.empid = empid  
    self.exp = experience
```

def displaydata(self):

```
    self.display()  
    print("Employee id:", self.empid)  
    print("Experience:", self.exp).
```

# VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)  
(Affiliated to Osmania University)  
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : PP LAB

Name K.S.T.Sivani

Roll No. 1602-21-733-052 Page No. 147

Output:

Name: Ram

Age: 25

Employee id: 123

Experience: 10

Name: data

Age: 22

Employee id: 134

Experience: 12

2) Write a program to compare two date objects:

class date:

```
def info(self):  
    self.date = int(input("Enter the date:"))  
    self.month = int(input("Enter the month:"))  
    self.year = int(input("Enter the year:"))
```

```
def display(self):  
    print("date:", self.date, "/", self.month, "/",  
          self.year)
```

```
def __gt__(self, b):  
    if self.year > b.year:  
        return True
```

```
else:  
    return False
```

# VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)  
(Affiliated to Osmania University)

Hyderabad - 500 031.

DEPARTMENT OF

: CSE

NAME OF THE LABORATORY : PPLAB

Name K.S.I.Sivami

Roll No. 1602-21-733-052 Page No. 148

```
d1 = date()
d2 = date()
d1.info()
d2.info()
d1.display()
d2.display()
if d1 > d2:
    print("d1 > d2")
else:
    print("d2 > d1")
```

Output:

Enter the date: 25

Enter the month: 08

Enter the year: 2003

Enter the date: 07

Enter the month: 01

Enter the year: 2022

date: 25/8/2003

date: 7/1/2022

d2 > d1

- 3) Program that has class point. Define another class location that points the reflection of destination on the x-axis.

\* class point():

```
def __init__(self):
    self.x = int(input("x-coordinate"))
    self.y = int(input("y-coordinate"))
```

class location(point):

```
def __init__(self):
    point.__init__(self)
```

def reflection(self):

```
    point(self.x, -self.y)
```

l = location()

l.reflection()

Output:

x-coordinate: 5

y-coordinate: 2

(5, -2)

DEPARTMENT OF

: CSE

NAME OF THE LABORATORY : PP LAB

Name K Sree Indira Divani Roll No. 1602-21-733-052 Page No. 149

4) Write a menu driven program to overload + =, -=, ==,  
>= & <= operations on class time :

class time:

def \_\_init\_\_(sey, t):

sey.t = t

def \_\_iadd\_\_(sey, c):

sey.t += c.t

def \_\_isub\_\_(sey, c):

sey.t -= c.t

def \_\_eq\_\_(sey, c):

if (sey.t == c.t):

print("equal")

def \_\_ge\_\_(sey, c):

print(sey.t >= c.t)

def \_\_le\_\_(sey, c):

print(sey.t <= c.t)

print("1.add 2.Sub 3.equal 4.ge 5.le")

x=int(input("choice"))

t=time(10)

l=time(10)

if x==1:

t+=1

elif x==2:

t-=1

elif  $x=3$ :

$t = l$

elif  $x=4$ :

$t \geq l$

elif  $x=5$ :

$t \leq l$  kast mit return

else:

pass kast mit return

For: steckt mit return

Output: 1. add 2. sub 3. equal 4. ge 5. le

1. add 2. sub 3. equal 4. ge 5. le

pass kast mit return

if  $b < b$

ibest

zwei Kettensymbole mit zwei zentralen Punkten  
durch Kettensymbole für Neigung und Achsenwert  
verbinden

zwei Kettensymbole mit zwei zentralen Punkten  
durch Kettensymbole für Neigung und Achsenwert  
verbinden

zwei Kettensymbole mit zwei zentralen Punkten  
durch Kettensymbole für Neigung und Achsenwert  
verbinden

(( $\text{steigung}-x$ ) kugel) fin =  $\mu \cdot p_2$

(( $\text{steigung}+v$ ) kugel) fin =  $\mu \cdot p_2$

outpur

zwei Kettensymbole mit zwei zentralen Punkten  
durch Kettensymbole für Neigung und Achsenwert  
verbinden

zwei Kettensymbole mit zwei zentralen Punkten  
durch Kettensymbole für Neigung und Achsenwert  
verbinden

(( $\mu \cdot p_2 - \text{neigung}$ ) kugel)

(( $\mu \cdot p_2 + \text{neigung}$ ) kugel)