														Internal Assessment Test I-Dec 2023			
USN																	
Subject:	**PYTHON PROGRAMMING LABORATORY**										Sub Code:	21CSL46					
Date:	05-09-2023		Duration:		120Min's		Sem/Sec:		IV Sem-A								
Note: Answer any Five Full Questions															Marks	CO	RBT
1 a.# **a) Write a python program to download the all XKCD comics**															100	CO1	L2

Algorithm:

[45 for Procedore]

Program:

```
import requests
import os
from bs4 import BeautifulSoup

# Set the URL of the first XKCD comic
url = 'https://xkcd.com/1/'

# Create a folder to store the comics
if not os.path.exists('untitled Folder 2'):
    os.makedirs('untitled Folder 2')

# Loop through all the comics
while True:
    # Download the page content
    res = requests.get(url)
    res.raise_for_status()

# Parse the page content using BeautifulSoup
    soup = BeautifulSoup(res.text, 'html.parser')

    # Find the URL of the comic image
    comic_elem = soup.select('#comic img')
    if comic_elem == []:
        print('Could not find comic image.')
    else:
        comic_url = 'https:' + comic_elem[0].get('src')
```

[45 Conducting program]

[10 Viva]

```
# Download the comic image
        print(f'Downloading {comic_url}...')
        res = requests.get(comic_url)
        res.raise_for_status()

# Save the comic image to the xkcd_comics folder
        image_file = open(os.path.join('untitled Folder
2', os.path.basename(comic_url)), 'wb')
        for chunk in res.iter_content(100000):
            image_file.write(chunk)
        image_file.close()
# Get the URL of the previous comic
    prev_link = soup.select('a[rel="prev"]')[0]
    if not prev_link:
        break
    url = 'https://xkcd.com' + prev_link.get('href')
print('All comics downloaded.')
```

Output:

```
Downloading https://imgs.xkcd.com/comics/barrel_cropped_(
1).jpg...
Downloading https://imgs.xkcd.com/comics/cosmological_nos
talgia_content.png...
Downloading https://imgs.xkcd.com/comics/linguistics_goss
ip.png...
```

b) Demonstrate python program to read the data from the spreadsheet and write the data in to the spreadsheet

Algorithm:

Program:

```
from openpyxl import Workbook
from openpyxl.styles import Font
wb = Workbook()
sheet = wb.active
sheet.title = "Language"
```

```
wb.create_sheet(title = "Capital")

lang = ["Kannada", "Telugu", "Tamil"]
state = ["Karnataka", "Telangana", "Tamil Nadu"]
capital = ["Bengaluru", "Hyderabad", "Chennai"]
code =['KA', 'TS', 'TN']

sheet.cell(row = 1, column = 1).value = "State"
sheet.cell(row = 1, column = 2).value = "Language"
sheet.cell(row = 1, column = 3).value = "Code"



ft = Font(bold=True)
for row in sheet["A1:C1"]:
    for cell in row:
        cell.font = ft

for i in range(2,5):
    sheet.cell(row = i, column = 1).value = state[i-2]
    sheet.cell(row = i, column = 2).value = lang[i-2]
    sheet.cell(row = i, column = 3).value = code[i-2]

wb.save("demo.xlsx")

sheet = wb["Capital"]
sheet.cell(row = 1, column = 1).value = "State"
sheet.cell(row = 1, column = 2).value = "Capital"
sheet.cell(row = 1, column = 3).value = "Code"

ft = Font(bold=True)
for row in sheet["A1:C1"]:
    for cell in row:
        cell.font = ft

for i in range(2,5):
```

```
    sheet.cell(row = i, column = 1).value = state[i-2]
    sheet.cell(row = i, column = 2).value = capital[i-2]
    sheet.cell(row = i, column = 3).value = code[i-2]

wb.save("demo.xlsx")

srchCode = input("Enter state code for finding capital ")
for i in range(2,5):
    data = sheet.cell(row = i, column = 3).value
    if data == srchCode:
        print("Corresponding capital for code", srchCode,
"is", sheet.cell(row = i, column = 2).value)

sheet = wb["Language"]

srchCode = input("Enter state code for finding language
")
for i in range(2,5):
    data = sheet.cell(row = i, column = 3).value
    if data == srchCode:
        print("Corresponding language for code",
srchCode, "is", sheet.cell(row = i, column = 2).value)

wb.close()
```

a) By using the concept of inheritance write a python program to find the area of triangle, circle and rectangle.

[45 for Procedore]

CO1

L2

Algorithm:

Program:

```
class shape():
    def area(self):
        raise NotImplementedError()
```

```
    def display(self):
        raise NotImplementedError()
#Implementation of Inheritance
class circle(shape):
    def __init__(self, radius):
        self.radius = Radius
        self.area_circle = 0
#calculate area of circle
    def area(self):
        self.area_circle = 3.142 * self.radius *
self.radius

    def display(self):
        print ("Area of Circle: ",self.area_circle)

class triangle(shape):
    def __init__(self, breadth, height):
        self.breadth = Breadth
        self.height = Height
        self.area_triangle = 0
#calculate area of triangle
    def area(self):
        self.area_triangle = 0.5 * self.breadth *
self.height

    def display(self):
        print ("Area of triangle: ",self.area_triangle)

class rectangle(shape):
    def __init__(self, length, breadth):
        self.length = Length
        self.breadth = Breadth
        self.area_rectangle = 0
#calculate area of rectangle
    def area(self):
        self.area_rectangle =  self.length *
self.breadth
```

[45 Conducting program]

[10 Viva]

```
    def display(self):
        print ("Area of rectangle: ",self.area_rectangle)
# user to enter the basic values to calculate the area
Radius = int(input("enter radius"))
Height = int(input("enter height"))
Breadth = int(input("enter base"))
Length = int(input("enter length"))
Width = int(input("enter width"))

cir_obj = circle(Radius)
cir_obj.area()
cir_obj.display()

tri_obj = triangle(Height,Breadth)
tri_obj.area()
tri_obj.display()

rect_obj = rectangle(Length,Width)
rect_obj.area()
rect_obj.display()
```

output :

```
enter radius5
enter height6
enter base7
enter length8
enter width9
Area of Circle:  78.55
Area of triangle:  21.0
Area of rectangle:  56
```

b) Write a python program by creating a class called Employee to store the details of Name, Employee_ID, Department and Salary, and implement a method to update salary of employees belonging to a given department

Algorithm:

Program:

```
class Employee:
    def __init__(self, emp_name, emp_id, emp_salary,
emp_department):
        self.emp_name = emp_name
        self.emp_id = emp_id
        self.emp_salary = emp_salary
        self.emp_department = emp_department
#Method to calculate the salary of employee
    def calculate_salary(self, emp_salary, hours_worked):
        overtime = 0
        if hours_worked > 50:
            overtime = hours_worked - 50
        self.emp_salary = self.emp_salary + (overtime *
(self.emp_salary / 50))

    def assign_department(self, emp_department):
        self.emp_department = emp_department

    def print_employee_details(self):
        print("\nName: ", self.emp_name)
        print("ID: ", self.emp_id)
        print("Salary: ", self.emp_salary)
        print("Department: ", self.emp_department)
        print("----------------------")



employee1 = Employee("ADAMS", "E7876", 50000,
"ACCOUNTING")
employee2 = Employee("JONES", "E7499", 45000, "RESEARCH")
employee3 = Employee("MARTIN", "E7900", 50000, "SALES")
employee4 = Employee("SMITH", "E7698", 55000,
"OPERATIONS")


print("Original Employee Details:")
```

```
employee1.print_employee_details()
employee2.print_employee_details()
employee3.print_employee_details()
employee4.print_employee_details()


# Change the departments of employee1 and employee4
employee1.assign_department("OPERATIONS")
employee4.assign_department("SALES")


# Now calculate the overtime of the employees who are eligible:
employee2.calculate_salary(45000, 52)
employee4.calculate_salary(45000, 60)


print("Updated Employee Details:")
employee1.print_employee_details()
employee2.print_employee_details()
employee3.print_employee_details()
employee4.print_employee_details(
```

	[45 writing] **[45 conducting]** [10]	50	CO1	L1