Dr. Ambedkar Institute of Technology

Bengaluru-56

Department of Computer Science and Engineering

Scheme and Syllabus – OBE - CBCS – 2022 -2023

<table>
<tr><td></td><td colspan="7"><b>PYTHON PROGRAMMING LABORATORY</b></td></tr>
<tr><td>Course Code</td><td colspan="7"><b>21CSL405</b></td></tr>
<tr><td>Category</td><td colspan="7">Engineering Science Course (ES)</td></tr>
<tr><td rowspan="3">Scheme and Credits</td><td colspan="5">No. of Hours/Week</td><td rowspan="2">Total Hrs./semester</td><td rowspan="2">Credits</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>SS</td><td>Total</td></tr>
<tr><td>0</td><td>0</td><td>2</td><td>0</td><td>2</td><td>24</td><td>1</td></tr>
<tr><td>CIE Marks: 50</td><td colspan="2">SEE Marks: 50</td><td colspan="2">Total Max. Marks: 100</td><td colspan="3">Duration of SEE: 03 Hours</td></tr>
</table>

Course objectives to:

1. Explain problem statements and identify appropriate solutions
2. Demonstrate the use of IDE, C Compiler, and identify and rectify the syntax and syntactic errors during programming.
3. Development of algorithms and programs using constructs of C programming language
4. Reporting the observations

Lab Programs	
Sl. No.	**PART A** List of problems for which student should develop program and execute in the Laboratory
1.	Aim: Introduce the Python fundamentals, data types, operators, flow control and exception handling in Python a. Write a python program to find the best of two test average marks out of three test's marks accepted from the user. b. Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number. Datatypes: https://www.youtube.com/watch?v=gCCVsvgR2KU Operators: https://www.youtube.com/watch?v=v5MR5JnKcZI Flow Control: https://www.youtube.com/watch?v=PqFKRqpHrjw For loop: https://www.youtube.com/watch?v=0ZvaDa8eT5s While loop: https://www.youtube.com/watch?v=HZARImviDxg Exceptions: https://www.youtube.com/watch?v=6SPDvPK38tw
2.	Aim: Demonstrating creation of functions, passing parameters and return values

<table>
<tr><td></td><td colspan="2">a. Defined as a function F as Fn = Fn-1 + Fn-2. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed.
b. Develop a python program to convert binary to decimal, octal to hexadecimal using functions.
Functions: https://www.youtube.com/watch?v=BVfCWuca9nw
Arguments: https://www.youtube.com/watch?v=ijXMGpoMkhQ
Return value: https://www.youtube.com/watch?v=nuNXiEDnM44</td></tr>
<tr><td><b>3.</b></td><td colspan="2">Aim: Demonstration of manipulation of strings using string methods
a. Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters.
b. Write a Python program to find the string similarity between two given strings
Sample Output:</td></tr>
<tr><td></td><td><b>Sample Output:</b>
Original string:
Python Exercises</td><td><b>Sample Output:</b>
Original string:
Python Exercises</td></tr>
</table>

<table>
<tr><td rowspan="2"></td><td>Python Exercises
Similarity between two said strings:
1.0</td><td>Python Exercise
Similarity between two said strings:
0.967741935483871</td></tr>
<tr><td colspan="2">Strings: https://www.youtube.com/watch?v=lSItwlnF0eU
String functions: https://www.youtube.com/watch?v=9a3CxJyTq00</td></tr>
<tr><td><b>4.</b></td><td colspan="2"><b>Aim:</b> Discuss different collections like list, tuple and dictionary
a. Write a python program to implement insertion sort and merge sort using lists
b. Write a program to convert roman numbers in to integer values using dictionaries.
Lists: https://www.youtube.com/watch?v=Eaz5e6M8tL4
List methods: https://www.youtube.com/watch?v=8-RDVWGktuI
Tuples: https://www.youtube.com/watch?v=bdS4dHIJGBc
Tuple operations: https://www.youtube.com/watch?v=TItKabcTTQ4
Dictionary: https://www.youtube.com/watch?v=4Q0pW8XBOkc
Dictionary methods: https://www.youtube.com/watch?v=oLeNHuORpNY</td></tr>
<tr><td><b>5.</b></td><td colspan="2"><b>Aim:</b> Demonstration of pattern recognition with and without using regular expressions
a. Write a function called isphonenumber () to recognize a pattern 415-555-4242 without using regular expression and also write the code to recognize the same pattern using regular expression.
b. Develop a python program that could search the text in a file for phone numbers (+919900889977) and email addresses (sample@gmail.com)</td></tr>
</table>

	Regular expressions: https://www.youtube.com/watch?v=LnzFnZfHLS4
6.	**Aim:** Demonstration of reading, writing and organizing files. a. Write a python program to accept a file name from the user and perform the following operations 1. Display the first N line of the file 2. Find the frequency of occurrence of the word accepted from the user in the file b. Write a python program to create a ZIP file of a particular folder which contains several files inside it. Files: https://www.youtube.com/watch?v=vuyb7CxZgbU https://www.youtube.com/watch?v=FqcjKewJTQ0 File organization: https://www.youtube.com/watch?v=MRuq3SRXses
7.	**Aim:** Demonstration of the concepts of classes, methods, objects and inheritance a. By using the concept of inheritance write a python program to find the area of triangle, circle and rectangle. 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. OOP's concepts: https://www.youtube.com/watch?v=qiSCMNBIP2g Inheritance: https://www.youtube.com/watch?v=Cn7AkDb4pIU
8.	**Aim:** Demonstration of classes and methods with polymorphism and overriding a. Write a python program to find the whether the given input is palindrome or not (for both string and integer) using the concept of

	polymorphism and inheritance. Overriding: https://www.youtube.com/watch?v=CcTzTuIsoFk
9.	**Aim:** Demonstration of working with excel spreadsheets and web scraping a. Write a python program to download the all XKCD comics b. Demonstrate python program to read the data from the spreadsheet and write the data in to the spreadsheet Web scraping: https://www.youtube.com/watch?v=ng2o98k983k Excel: https://www.youtube.com/watch?v=nsKNPHJ9iPc
10.	**Aim:** Demonstration of working with PDF, word and JSON files a. Write a python program to combine select pages from many PDFs b. Write a python program to fetch current weather data from the JSON file PDFs: https://www.youtube.com/watch?v=q70xzDG6nls https://www.youtube.com/watch?v=JhQVD7Y1bsA https://www.youtube.com/watch?v=FcrW-ESdY-A Word files: https://www.youtube.com/watch?v=ZU3cSl51jWE JSON files: https://www.youtube.com/watch?v=9N6a-VLBa2I
	PART B – Practical Based Learning
	A problem statement for each batch is to be generated in consultation with the co-examiner and student should develop an algorithm, program and execute the program for the given problem with appropriate outputs.
	Course Outcomes: **CO1:** Demonstrate proficiency in handling of loops and creation of functions.

	CO2: Identify the methods to create and manipulate lists, tuples and dictionaries. **CO3:** Discover the commonly used operations involving regular expressions and file system. **CO4:** Interpret the concepts of Object-Oriented Programming as used in Python. **CO5:** Determine the need for scraping websites and working with PDF, JSON and other file formats.
	Suggested Learning Resources: 1. Gowrishankar S, Veena A, **“Introduction to Python Programming”**, 1st Edition, CRC Press/Taylor & Francis, 2018. ISBN-13: 978-0815394372 2. Al Sweigart, **“Automate the Boring Stuff with Python”,** 1st Edition, No Starch Press, 2015. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/) 3. Allen B. Downey, **“Think Python: How to Think Like a Computer Scientist”,** 2nd Edition, Green Tea Press, 2015. (Available under CC-BY-NC license at http://greenteapress.com/thinkpython2/thinkpython2.pdf)

MAPPING of COs with POs

CO-PO Mapping	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	**1**	**1**	**2**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
CO2	1	1	2	2	2	-	-	-	-	-	-	-

CO3	2	2	3	1	3	-	-	-	-	-	-	-
CO4	3	3	3	3	3	-	-	-	-	-	-	-
CO5	1	1	2	2	2	-	-	-	-	-	-	-

1a. Write a python program to find the best of two test average marks out of three test’s marks accepted from the user.

```
m1 = int (input("Enter the marks in the first test: "))
m2 = int (input("Enter the marks in second test: "))
m3 = int (input("Enter the marks in third test: "))
if (m1 > m2):
    if (m2 > m3):
        total = m1 + m2
    else:
        total = m1 + m3
elif (m1 > m3):
    total = m1 + m2
else:
    total = m2 + m3
Avg = total / 2
print ("The average of the best two test marks is: ",Avg)
```

Output:

Case 1:

```
Enter the marks in the first test: 20

Enter the marks in the second test: 15

Enter the marks in the third test: 22
```

The average of the best two test marks is: 21.0

Case 2:

Enter the marks in the first test: 20

Enter the marks in the second test: 23

Enter the marks in the third test: 18

The average of the best two test marks is: 21.5

OR

```
m1 = int(input("Enter marks for test1 : "))
m2 = int(input("Enter marks for test2 : "))
m3 = int(input("Enter marks for test3 : "))

if m1 <= m2 and m1 <= m3:
    Avg = (m2+m3)/2
elif m2 <= m1 and m2 <= m3:
    Avg = (m1+m3)/2
elif m3 <= m1 and m2 <= m2:
    Avg = (m1+m2)/2

print("Average of best two test marks  out of three test’s marks is",
Avg);
```

1b. Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number

```
Number = int(input("Please Enter any Number: "))
Reverse = 0
temp=Number
```

```
while(Number > 0):
        Reminder = Number %10
        Reverse = (Reverse *10) + Reminder
        Number = Number //10
print("\n Reverse of entered number is",Reverse)
if  temp == Reverse:
        print("The given number {0} is palindrome".format(temp))
else:
        print("The given number {0} is not palindrome".format(temp))
```

Output:

Case1:

```
Please Enter any Number: 121

Reverse of entered number is 121
The given number 121 is palindrome
```

Case2:

```
Please Enter any Number: 123

Reverse of entered number is 321
The given number 123 is not palindrome
```

```
Number = int(input("Please Enter any Number: "))
digit = int(input("Please Enter digit to find the occurrence: "))
count=0
while (Number > 0):

       # check if the digit is D         #n=1231 d=1
       if(Number % 10 == digit):     #1231%10=1  1==1
123%10=3  3==1     12%10=2
          count=count+1                                          #count=0+1=1
        Number = Number // 10        # num=1231//10=123
123//10=12
```

```
print(" digit {0} occurs {1} times ".format(digit,count))
```

Output
Case1:

```
Please Enter any Number: 123121
Please Enter digit to find the occurrence: 1
digit 1 occurs 3 times
```

Case2:

```
Please Enter any Number: 12277189292777456
Please Enter digit to find the occurrence: 7
digit 7 occurs 5 times
```

```
val = int(input("Enter a value : "))
str_val = str(val)
if str_val == str_val[::-1]:
    print("Palindrome")
else:
    print("Not Palindrome")

for i in range(10):
    if str_val.count(str(i)) > 0:
        print(str(i),"appears", str_val.count(str(i)), "times");
```

2a. Defined as a function F as Fn = Fn-1 + Fn-2. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed.

In mathematical terms, the sequence Fn of Fibonacci numbers is defined by the recurrence relation

$F_n = F_{n-1} + F_{n-2}$
With seed values

$F_0 = 0$ and $F_1 = 1$.

```
# Function for nth fibonacci number
# Taking 1st two fibonacci numbers as 0 and 1
#Fibonacci of a number without using recursion

def fibonacci(n):
    a = 0
    b = 1
    if n < 0:
        print("Incorrect input")
    elif n == 0:
        return a
    elif n == 1:
        return b
    else:
        for i in range(2, n):
            c = a + b
            a = b
            b = c
      return b

num = int(input("Enter a number : "))
print(fibonacci(num))
```

```
# Function for nth Fibonacci number
#Fibonacci of a number using recursion

def Fibonacci(n):
        if n<= 0:
                print("Incorrect input")
        # First Fibonacci number is 0
        elif n == 1:
                return 0
        # Second Fibonacci number is 1
        elif n == 2:
                return 1
        else:
                return Fibonacci(n-1)+Fibonacci(n-2)

num = int(input("Enter a number : "))
print(Fibonacci(num))
```

2b. Develop a python program to convert binary to decimal, octal to hexadecimal using functions

```
# Function calculates the decimal equivalent
# to given binary number

 def binaryToDecimal(binary):

    decimal, i = 0, 0
    while(binary != 0):      1010
        dec = binary % 10                          dec= 1%10=1
        decimal = decimal + dec * pow(2, i)
#decimal=2+1*pow(2,3)=10
        binary = binary//10                    bin=0//10=0
        i += 1                                 i=4
    return (decimal)
num = int(input("Enter a binary number : "))
print(binaryToDecimal (num))
```

#octal to hexadecimal

```
def oct2Hex(val):     10
    rev=val[::-1] - 01
    dec = 0
    i = 0
    for dig in rev:        01
        dec += int(dig) * 8**i     dec=0+1*8** 1=8
        i += 1                          i=2
    list=[]                    -empty list
    while dec != 0:                 8!=0
        list.append(dec%16)           list=[8]
        dec = dec // 16   dec=8//16=0

    nl=[]
    for elem in list[::-1]:      [8]
        if elem <= 9:
            nl.append(str(elem))
        else:
            nl.append(chr(ord('A') + (elem -10)))
    hex = "".join(nl)

    return hex

num = int(input("Enter a octal number : "))
print(oct2Hex(num))
```

3a. Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters.

```
sentence = input("Enter a sentence : ")        # s="Dr AIT Bangalore"
                                              #s.split("  ")  ["Dr" "AIT"  ]
wordList = sentence.split(" ")
print("This sentence has", len(wordList), "words")
```

```
digCnt = upCnt = loCnt = 0

for ch in sentence:
    if '0' <= ch <= '9':          ch>=’0’ and ch<=’9’
        digCnt += 1
    elif 'A' <= ch <= 'Z':
        upCnt += 1
    elif 'a' <= ch <= 'z':
        loCnt += 1

print("This sentence has", digCnt, "digits", upCnt, "upper case letters", 
loCnt, "lower case letters")
```

Output

```
Enter a sentence : Rama went to Devaraja market to pick 2 kgs of 
vegetable
This sentence has 11 words
This sentence has 1 digits 2 upper case letters 42 lower case letters
```

3b. String Similarity

Write a Python program to find the string similarity between two given strings.

```
str1 = input("Enter String 1 \n")
str2 = input("Enter String 2 \n")

if len(str2) < len(str1):     str1-python- len(str1)-6   str2-pytho-5
    short = len(str2)-5
    long = len(str1)-6
else:
    short = len(str1)
    long = len(str2)

matchCnt = 0
for i in range(short):      for i in range 5
    if str1[i] == str2[i]:      st1r[0]== str2[0]
        matchCnt += 1          m=5

print("Similarity between two said strings:")
print(matchCnt/long)     5/6=
```

Output

```
Enter String 1
Python Exercises
Enter String 2
Python Exercises
Similarity between two said strings:
1.0
Enter String 1
Python Exercises
Enter String 2
Python Exercise
Similarity between two said strings:
0.9375
```

4a. Write a python program to implement insertion sort and merge sort using lists.

```
import random
def merge_sort(lst):
    if len(lst) > 1:
        mid = len(lst) // 2
        left_half = lst[:mid]
        right_half = lst[mid:]

        merge_sort(left_half)
        merge_sort(right_half)                                      12 10  14  9

        i = j = k = 0                               12  10       14  9
                                                     12       10       14       9
                                                       10 12            9 14             9 10 12 14



        while i < len(left_half) and j < len(right_half):
            if left_half[i] < right_half[j]:
                lst[k] = left_half[i]
                i += 1
            else:
                lst[k] = right_half[j]
                j += 1
            k += 1

        while i < len(left_half):
            lst[k] = left_half[i]
            i += 1
            k += 1

        while j < len(right_half):
            lst[k] = right_half[j]
            j += 1
            k += 1

    return lst
12 10 14  9
10 12 14 9
```

10 12 9 14

9 10 12 14

```
def insertion_sort(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1
        while j >= 0 and key < arr[j]:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key

my_list = []

for i in range(10):
    my_list.append(random.randint(0, 999))

print("\nUnsorted List")
print(my_list)
print("Sorting using Insertion Sort")
insertion_sort(my_list)
print(my_list)


my_list = []

for i in range(10):
    my_list.append(random.randint(0, 999))

print("\nUnsorted List")
print(my_list)
print("Sorting using Merge Sort")
merge_sort(my_list)
```

```
print(my_list)
```

Output

```
Unsorted List
[932, 111, 226, 685, 543, 589, 918, 539, 294, 717]
Sorting using Insertion Sort
[111, 226, 294, 539, 543, 589, 685, 717, 918, 932]

Unsorted List
[613, 176, 828, 265, 65, 326, 359, 919, 514, 868]
Sorting using Merge Sort
[65, 176, 265, 326, 359, 514, 613, 828, 868, 919]
```

Roman to Integer Conversion

4b. Write a program to convert roman numbers in to integer values using dictionaries.

```
def roman2Dec(romStr):      XVII-17
    roman_dict ={'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}
    # Analyze string backwards
    romanBack = list(romStr)[::-1]    - IIVX
    value = 0
    # To keep track of order
    rightVal = roman_dict[romanBack[0]]    rightval=1
    for numeral in romanBack:                    IIVX
        leftVal = roman_dict[numeral]    leftval=10
        # Check for subtraction
        if leftVal < rightVal:          if 10<1
            value -= leftVal
        else:
             value += leftVal     value=1 +1=2+5=7+10=17
        rightVal = leftVal    right=10
    return value
```

```
romanStr = input("Enter a Roman Number : ")
print(roman2Dec(romanStr))
Output
Enter a Roman Number : XVII
17

Enter a Roman Number : MLXVI
1066
```

Check Phone Number

5a. Write a function called isphonenumber () to recognize a pattern 415-555-4242 without using regular expression and also write the code to recognize the same pattern using regular expression.

```
import re

def isphonenumber(numStr):
    if len(numStr) != 12:
        return False
    for i in range(len(numStr)):
        if i==3 or i==7:
            if numStr[i] != "-":
                return False
        else:
            if numStr[i].isdigit() == False:
                return False
    return True

def chkphonenumber(numStr):
    ph_no_pattern = re.compile(r'^\d{3}-\d{3}-\d{4}$')
    if ph_no_pattern.match(numStr):
        return True
    else:
        return False

ph_num = input("Enter a phone number : ")
print("Without using Regular Expression")
if isphonenumber(ph_num):
    print("Valid phone number")
else:
    print("Invalid phone number")

print("Using Regular Expression")
if chkphonenumber(ph_num):
    print("Valid phone number")
else:
    print("Invalid phone number")
```

Output

```
Enter a phone number : 444-654-5656
Without using Regular Expression
Valid phone number
Using Regular Expression
Valid phone number
Enter a phone number : 45A4-444-878
Without using Regular Expression
Invalid phone number
Using Regular Expression
Invalid phone number
```

Search Phone Number & Email

5b. Develop a python program that could search the text in a file for phone numbers (+919900889977) and email addresses (sample@gmail.com)

```
import re

# Define the regular expression for phone numbers
phone_regex = re.compile(r'\+\d{12}')
email_regex = re.compile(r'[A-Za-z0-9._]+@[A-Za-z0-9]+\.[A-Z|a-z]{2,}')
# Open the file for reading
with open('example.txt', 'r') as f:
    # Loop through each line in the file
    for line in f:
        # Search for phone numbers in the line
        matches = phone_regex.findall(line)
        # Print any matches found
        for match in matches:
            print(match)

        matches = email_regex.findall(line)
        # Print any matches found
        for match in matches:
            print(match)
```

Output

```
+918151894220
+829392938876
+918768456234
prakash81.82@gmail.in
```

Question 6

File Operations

6a. Write a python program to accept a file name from the user and perform the following operations

Display the first N line of the file

Find the frequency of occurrence of the word accepted from the user in the file

```
import os.path
import sys

fname = input("Enter the filename : ")

if not os.path.isfile(fname):
    print("File", fname, "doesn't exists")
    sys.exit(0)

infile = open(fname, "r")

lineList = infile.readlines()

for i in range(20):
    print(i+1, ":", lineList[i])

word = input("Enter a word : ")
cnt = 0
for line in lineList:
    cnt += line.count(word)

print("The word", word, "appears", cnt, "times in the file")
```

Output

```
Enter the filename : example.txt
1 : this is phone number +918151894220
2 : no phone number here
3 : here we have one +829392938876
```

```
4 : we have an email prakash81.82@gmail.in and a number
+918768456234
5 : nothing of that sort here
6 : Better hope the life-inspector doesn't come around while you have
your
7 : life in such a mess.
8 : You can create your own opportunities this week.  Blackmail a senior
executive.
9 : Be different: conform.
10 : Be cheerful while you are alive.
11 :          -- Phathotep, 24th Century B.C.
12 : Q:      How many journalists does it take to screw in a light bulb?
13 : A:      Three.  One to report it as an inspired government program to
bring
14 :   light to the people, one to report it as a diabolical government plot
15 :   to deprive the poor of darkness, and one to win a Pulitzer prize for
16 :   reporting that Electric Company hired a light bulb-assassin to
break
17 :   the bulb in the first place.
18 : Q:      Why did the astrophysicist order three hamburgers?
19 : A:      Because he was hungry.
20 : Q:      Why haven't you graduated yet?
Enter a word : the
The word the appears 7 times in the file
```

Zip operation on a folder

6b. Develop a program to backing Up a given Folder (Folder in a current working directory) into a ZIP File by using relevant modules and suitable methods.

```
import os
import sys
import pathlib
import zipfile

dirName = input("Enter Directory name that you want to backup : ")

if not os.path.isdir(dirName):
   print("Directory", dirName, "doesn't exists")
   sys.exit(0)

curDirectory = pathlib.Path(dirName)

with zipfile.ZipFile("myZip.zip", mode="w") as archive:
   for file_path in curDirectory.rglob("*"):
      archive.write(file_path,
arcname=file_path.relative_to(curDirectory))

if os.path.isfile("myZip.zip"):
   print("Archive", "myZip.zip", "created successfully")
else:
   print("Error in creating zip archive")
```

Output

```
Enter Directory name that you want to backup : zipDemo
Archive myZip.zip created successfully
```

Question 7
Inheritance
7a. By using the concept of inheritance write a python program to find the area of triangle, circle and rectangle.

```
import math
class Shape:
   def __init__(self):
        self.area = 0
        self.name = ""

    def showArea(self):
        print("The area of the", self.name, "is", self.area, "units")

class Circle(Shape):
    def __init__(self,radius):
        self.area = 0
        self.name = "Circle"
        self.radius = radius

    def calcArea(self):
        self.area = math.pi * self.radius * self.radius

class Rectangle(Shape):
    def __init__(self,length,breadth):
        self.area = 0
        self.name = "Rectangle"
        self.length = length
        self.breadth = breadth

    def calcArea(self):
        self.area = self.length * self.breadth

class Triangle(Shape):
    def __init__(self,base,height):
        self.area = 0
        self.name = "Triangle"
        self.base = base
        self.height = height
    def calcArea(self):
        self.area = self.base * self.height / 2



c1 = Circle(5)
c1.calcArea()
c1.showArea()


r1 = Rectangle(5, 4)
r1.calcArea()
r1.showArea()


t1 = Triangle(3, 4)
t1.calcArea()
t1.showArea()
```

Output

```
The area of the Circle is 78.53981633974483 units
The area of the Rectangle is 20 units
The area of the Triangle is 6.0 units
```

Employee Details

7b. 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.

```
class Employee:
    def __init__(self):
        self.name = ""
        self.empId = ""
        self.dept = ""
        self.salary = 0

    def getEmpDetails(self):
        self.name = input("Enter Employee name : ")
        self.empId = input("Enter Employee ID : ")
        self.dept = input("Enter Employee Dept : ")
        self.salary = int(input("Enter Employee Salary : "))

    def showEmpDetails(self):
        print("Employee Details")
        print("Name : ", self.name)
        print("ID : ", self.empId)
        print("Dept : ", self.dept)
        print("Salary : ", self.salary)

    def updtSalary(self):
        self.salary = int(input("Enter new Salary : "))
        print("Updated Salary", self.salary)


e1 = Employee()
e1.getEmpDetails()
e1.showEmpDetails()
e1.updtSalary()
```

Output

```
Enter Employee name : Sameer
Enter Employee ID : A123
Enter Employee Dept : CSE
Enter Employee Salary : 85750
```

```
Employee Details
Name :  Sameer
ID :  A123
Dept :  CSE
Salary :  85750

Enter new Salary : 88800
Updated Salary 88800
```

Question 8

Polymorphism and Inheritance

8a. Write a python program to find the whether the given input is palindrome or not (for both string and integer) using the concept of polymorphism and inheritance.

```
class PaliStr:
    def __init__(self):
        self.isPali = False

    def chkPalindrome(self, myStr):
        if myStr == myStr[::-1]:
            self.isPali = True
        else:
            self.isPali = False

        return self.isPali
```

```
class PaliInt(PaliStr):
    def __init__(self):
        self.isPali = False

    def chkPalindrome(self, val):
        temp = val
        rev = 0
        while temp != 0:
            dig = temp % 10
            rev = (rev*10) + dig
            temp = temp //10

        if val == rev:
            self.isPali = True
        else:
            self.isPali = False

        return self.isPali

st = input("Enter a string : ")

stObj = PaliStr()
if stObj.chkPalindrome(st):
    print("Given string is a Palindrome")
else:
    print("Given string is not a Palindrome")

val = int(input("Enter a integer : "))

intObj = PaliInt()
if intObj.chkPalindrome(val):
    print("Given integer is a Palindrome")
```

```
else:
    print("Given integer is not a Palindrome")
```

Output

```
Enter a string : madam
Given string is a Palindrome
Enter a integer : 567587
Given integer is not a Palindrome
Enter a string : INDIA
Given string is not a Palindrome
Enter a integer : 6789876
Given integer is a Palindrome
```

Question 9
Download XKCD comics
9a. Write a python program to download the all XKCD comics

```
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('xkcd_comics'):
    os.makedirs('xkcd_comics')

# 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')

        # 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('xkcd_comics',
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/radians_are_cursed.png...
Downloading
https://imgs.xkcd.com/comics/presents_for_biologists.png...
Downloading https://imgs.xkcd.com/comics/launch_window.png...
Downloading https://imgs.xkcd.com/comics/obituary_editor.png...
Downloading https://imgs.xkcd.com/comics/fanservice.png...
Downloading https://imgs.xkcd.com/comics/hand_dryers.png...
```

Spreadsheet Operations

9b. Demonstrate python program to read the data from the spreadsheet and write the data in to the spreadsheet

```
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()
```

Output

```
Enter state code for finding capital KA
Corresponding capital for code KA is Bengaluru
Enter state code for finding language TS
Corresponding language for code TS is Telugu
```

Question 10
Merge selected pages from Multiple PDFs to a new PDF
Write a python program to combine select pages from many PDFs

```
from PyPDF2 import PdfWriter, PdfReader

num = int(input("Enter page number you want combine from multiple
documents "))
```

```
pdf1 = open('birds.pdf', 'rb')
pdf2 = open('birdspic.pdf', 'rb')

pdf_writer = PdfWriter()

pdf1_reader = PdfReader(pdf1)
page = pdf1_reader.pages[num - 1]
pdf_writer.add_page(page)

pdf2_reader = PdfReader(pdf2)
page = pdf2_reader.pages[num - 1]
pdf_writer.add_page(page)

with open('output.pdf', 'wb') as output:
    pdf_writer.write(output)
```

Output

This program allows you to extract specific pages from two PDF files, “birds.pdf” and “birdspic.pdf,” by entering the page numbers as user input. Once you input the desired page numbers, the program fetches those pages from both PDF files and combines them into a new file called “output.pdf.” This way, you can easily compile the desired pages from multiple PDF files into one document for your convenience.

```
Enter page number you want combine from multiple documents 3
birdsDownload
birdspicDownload
outputDownload
```

Fetch weather data from the JSON

10b.Write a python program to fetch current weather data from the JSON file

```
import json

# Load the JSON data from file
with open('weather_data.json') as f:
    data = json.load(f)

# Extract the required weather data
current_temp = data['main']['temp']
humidity = data['main']['humidity']
weather_desc = data['weather'][0]['description']

# Display the weather data
print(f"Current temperature: {current_temp}°C")
print(f"Humidity: {humidity}%")
print(f"Weather description: {weather_desc}")
```

JSON File :

```
{
  "coord": {
    "lon": -73.99,
    "lat": 40.73
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "clear sky",
      "icon": "01d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 15.45,
    "feels_like": 12.74,
    "temp_min": 14.44,
    "temp_max": 16.11,
    "pressure": 1017,
    "humidity": 64
  },
  "visibility": 10000,
  "wind": {
    "speed": 4.63,
    "deg": 180
  },
  "clouds": {
    "all": 1
  },
  "dt": 1617979985,
  "sys": {
    "type": 1,
    "id": 5141,
    "country": "US",
    "sunrise": 1617951158,
    "sunset": 1618000213
  },
  "timezone": -14400,
  "id": 5128581,
  "name": "New York",
  "cod": 200
}
```

Output

```
Current temperature: 15.45°C
Humidity: 64%
```

Weather description: clear sky