NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE

(NAAC Accredited)

(Approved by AICTE , Affiliated to APJ Abdul Kalam Technological University, Kerala)

Pampady, Thiruvilwamala(PO), Thrissur(DT), Kerala 680 588

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB MANUAL

CS110 COMPUTER SCIENCE WORKSHOP

VISION OF THE INSTITUTION

To mould true citizens who are millennium leaders and catalysts of change through excellence in education.

MISSION OF THE INSTITUTION

NCERC is committed to transform itself into a center of excellence in Learning and Research in Engineering and Frontier Technology and to impart quality education to mould technically competent citizens with moral integrity, social commitment and ethical values.

We intend to facilitate our students to assimilate the latest technological know-how and to imbibe discipline, culture and spiritually, and to mould them in to technological giants, dedicated research scientists and intellectual leaders of the country who can spread the beams of light and happiness among the poor and the underprivileged.

ABOUT THE DEPARTMENT

- ♦ Established in: 2002
- ♦ Course offered : B.Tech in Computer Science and Engineering

  M.Tech in Computer Science and Engineering

  M.Tech in Cyber Security

- ♦ Approved by AICTE New Delhi and Accredited by NAAC
- ♦ Certified by ISO 9001-2015
- ♦ Affiliated to A P J Abdul Kalam Technological University, Kerala.

DEPARTMENT VISSION

Producing Highly Competent, Innovative and Ethical Computer Science and Engineering Professionals to facilitate continuous technological advancement.

DEPARTMENT MISSION

1. To Impart Quality Education by creative Teaching Learning Process
2. To Promote cutting-edge Research and Development Process to solve real world problems with emerging technologies.
3. To Inculcate Entrepreneurship Skills among Students.
4. To cultivate Moral and Ethical Values in their Profession.

PROGRAMME EDUCATIONAL OBJECTIVES

PEO1: Graduates will be able to Work and Contribute in the domains of Computer Science and Engineering through lifelong learning.

PEO2: Graduates will be able to Analyse, design and development of novel Software Packages, Web Services, System Tools and Components as per needs and specifications.

PEO3: Graduates will be able to demonstrate their ability to adapt to a rapidly changing environment by learning and applying new technologies.

PEO4: Graduates will be able to adopt ethical attitudes, exhibit effective communication skills, Team work and leadership qualities.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO)

PSO1: Ability to Formulate and Simulate Innovative Ideas to provide software solutions for Real-time Problems and to investigate for its future scope.

PSO2: Ability to learn and apply various methodologies for facilitating development of high quality System Software Tools and Efficient Web Design Models with a focus on performance optimization.

PSO3: Ability to inculcate the Knowledge for developing Codes and integrating hardware/software products in the domains of Big Data Analytics, Web Applications and Mobile Apps to create innovative career path and for the socially relevant issues.

COURSE OUTCOME

CO 1	To identify common hardware components and their purpose
CO 2	To implement the algorithms studied in introduction to computing and problem solving course
CO 3	To implement control structures, Iterations, Recursive functions, Lists, Tuples and Dictionaries
CO 4	To implement the operations of files
CO 5	To gain sufficient awareness about latest software tools
CO 6	To develop the programs in python for common problems of reasonable complexity

MAPPING OF COURSE OUTCOMES WITH PROGRAM OUTCOMES

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	3		3		3							3
CO 2	3	3	3	3								3
CO 3	3	3	3	3								3
CO 4	3	3	3	3	3							3
CO 5	3			3	3							3
CO 6	3	3	3	3								3

MAPPING OF COURSE OUTCOMES WITH PROGRAM SPECIFIC OUTCOMES

	PSO1	PSO2	PSO3
CO1	3		
CO2	3	3	3
CO3		3	3
CO4		3	3
CO5	3	3	3
CO6		2	3

Note: H-Highly correlated=3, M-Medium correlated=2, L-Less correlated=1

PREPARATION FOR THE LABORATORY SESSION
GENERAL INSTRUCTIONS TO STUDENTS

1. Read carefully and understand the description of the experiment in the lab manual. You may go to the lab at an earlier date to look at the experimental facility and understand it better. Consult the appropriate references to be completely familiar with the concepts and hardware.

2. Make sure that your observation for previous week experiment is evaluated by the faculty member and you have transferred all the contents to your record before entering to the lab/workshop.

3. At the beginning of the class, if the faculty or the instructor finds that a student is not adequately prepared, they will be marked as absent and not be allowed to perform the experiment.

4. Bring necessary material needed (writing materials, graphs, calculators, etc.) to perform the required preliminary analysis. It is a good idea to do sample calculations and as much of the analysis as possible during the session. Faculty help will be available. Errors in the procedure may thus be easily detected and rectified.

5. Please actively participate in class and don’t hesitate to ask questions. Please utilize the teaching assistants fully. To encourage you to be prepared and to read the lab manual before coming to the laboratory, unannounced questions may be asked at any time during the lab.

6. Carelessness in personal conduct or in handling equipment may result in serious injury to the individual or the equipment. Do not run near moving machinery/equipment. Always be on the alert for strange sounds. Guard against entangling clothes in moving parts of machinery.

7. Students must follow the proper dress code inside the laboratory. To protect clothing from dirt, wear a lab coat. Long hair should be tied back. Shoes covering the whole foot will have to be worn.

8. In performing the experiments, please proceed carefully to minimize any water spills, especially on the electric circuits and wire.

9. Maintain silence, order and discipline inside the lab. Don’t use cell phones inside the laboratory.

10. Any injury no matter how small must be reported to the instructor immediately.

11. Check with faculty members one week before the experiment to make sure that you have the handout for that experiment and all the apparatus.

AFTER THE LABORATORY SESSION

1. Clean up your work area.

2. Check with the technician before you leave.

3. Make sure you understand what kind of report is to be prepared and due submission of record is next lab class.

4. Do sample calculations and some preliminary work to verify that the experiment was successful

MAKE-UPS AND LATE WORK

Students must participate in all laboratory exercises as scheduled. They must obtain permission from the faculty member for absence, which would be granted only under justifiable circumstances. In such an event, a student must make arrangements for a make-up laboratory, which will be scheduled when the time is available after completing one cycle. Late submission will be awarded less mark for record and internals and zero in worst cases.

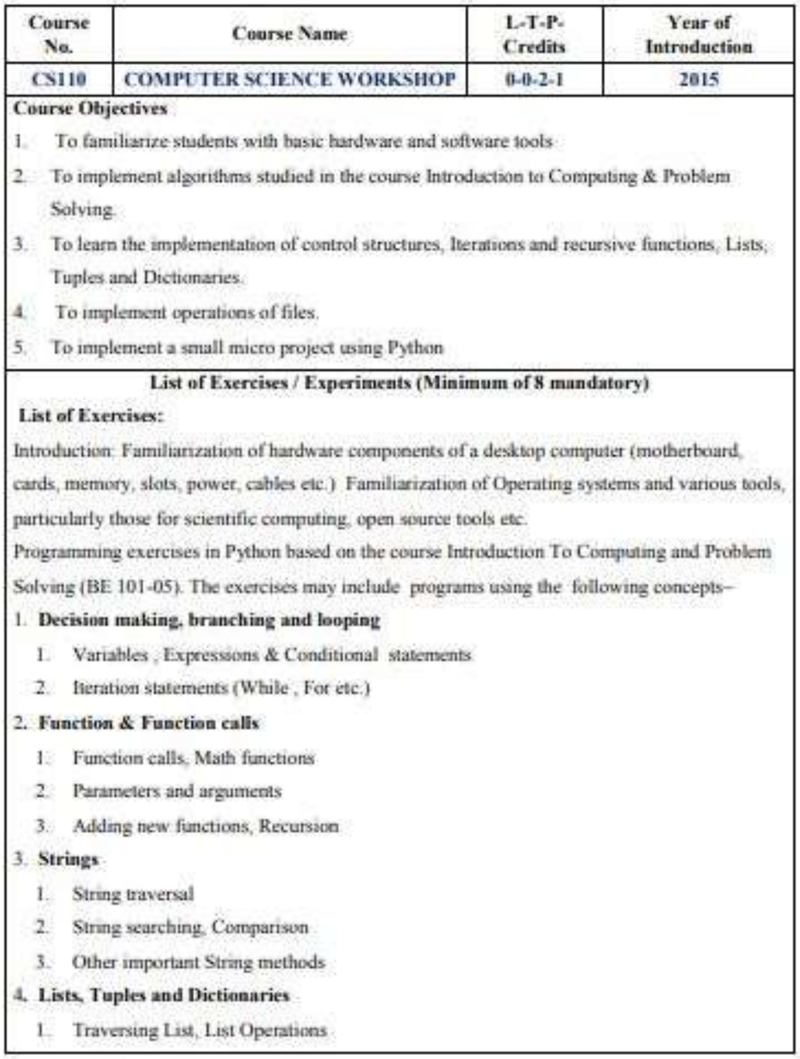
LABORATORY POLICIES

1. Food, beverages & mobile phones are not allowed in the laboratory at any time.

2. Do not sit or place anything on instrument benches.

3. Organizing laboratory experiments requires the help of laboratory technicians and staff. Be punctual.

SYLLABUS



Course No.	Course Name	L-T-P-Credits	Year of Introduction
CS110	COMPUTER SCIENCE WORKSHOP	0-0-2-1	2015

Course Objectives

1. To familiarize students with basic hardware and software tools
2. To implement algorithms studied in the course Introduction to Computing & Problem Solving.
3. To learn the implementation of control structures, Iterations and recursive functions, Lists, Tuples and Dictionaries.
4. To implement operations of files.
5. To implement a small micro project using Python

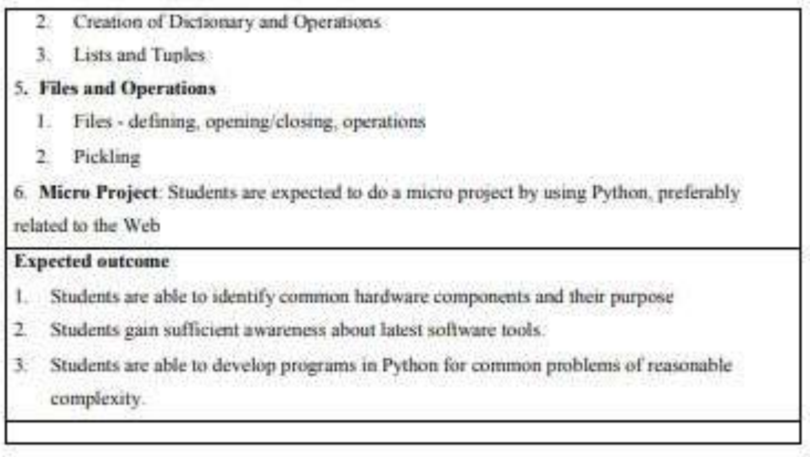
List of Exercises / Experiments (Minimum of 8 mandatory)

List of Exercises:

Introduction: Familiarization of hardware components of a desktop computer (motherboard, cards, memory, slots, power, cables etc.) Familiarization of Operating systems and various tools, particularly those for scientific computing, open source tools etc.

Programming exercises in Python based on the course Introduction To Computing and Problem Solving (BE 101-05). The exercises may include programs using the following concepts–

1. **Decision making, branching and looping**
    1. Variables , Expressions & Conditional statements
    2. Iteration statements (While , For etc.)
2. **Function & Function calls**
    1. Function calls, Math functions
    2. Parameters and arguments
    3. Adding new functions, Recursion
3. **Strings**
    1. String traversal
    2. String searching, Comparison
    3. Other important String methods
4. **Lists, Tuples and Dictionaries**
    1. Traversing List, List Operations



2. Creation of Dictionary and Operations
3. Lists and Tuples

5. **Files and Operations**

1. Files - defining, opening/closing, operations
2. Pickling

6. **Micro Project**: Students are expected to do a micro project by using Python, preferably related to the Web

Expected outcome

1. Students are able to identify common hardware components and their purpose
2. Students gain sufficient awareness about latest software tools.
3. Students are able to develop programs in Python for common problems of reasonable complexity.

INDEX

EXPERIMENT – 1
FAMILIARIZATION OF HARDWARE COMPONENTS OF A DESKTOP COMPUTER

OBJECTIVE:

To familiarize the different hardware components of a desktop computer.

COMPONENTS:

Hardware is the physical parts which makes the computer a machine. The different hardware components are as follows:

Monitor: It is the most common output device used with the computer. This is a TV like device that display the information from the computer to the user.

Keyboard – The keyboard is used to type something or input information to the computer. There are different designs and models of the keyboards in the market. The most common layout of the keyboard is QWERTY layout. A standard keyboard has 101 keys and embedded keys.

Mouse : Mouse is an input device. It is a pointing device that fits under the palm of the hand. Generally a mouse has two buttons left and right to perform different functions. One type of the mouse has a round ball under the bottom. Another type of the mouse use optical system to track the movement of the mouse.

Mother board: All electronic components in PC are mounted on a printed circuit board (PCB) called mother board. **Motherboard** is the computer's **main circuit board.** It's a thin plate that holds the processor, memory, connectors for the hard drive and optical drives, expansion cards to control the video and audio, and connections to your computer's ports (such as USB ports). The motherboard connects directly or indirectly to every part of the computer.

Processors: The main component of any motherboard is the microprocessors or central processing unit which controls the inner functions of the system. It functions as the brain of PC. The processor is usually inserted into the socket and is not soldered onto the motherboard. The common processors are 8086, Pentium III, Pentium 4 etc.

RAM chips: It is the Random Access Memory. Whenever the computer performs calculations, it temporarily stores the data in the RAM until it is needed.It is volatile.So when the computer is switched off, contents of RAM are lost.

ROM chips: They have data written on them at the time of manufacture. It is non-volatile. So the data is not lost after switching off PC. Contents of ROM cannot be deleted or altered.

BIOS: The Basic Input Output System, usually referred to as BIOS, is software stored on a small memory chip on themotherboard.BIOS instructs the computer on how to perform a number of basic functions such asbooting and keyboard control. In pc, BIOS contains all the code required to control the keyboard, display screen, disk drives and serial communications.

Real Time Clock Circuit: Real Time Clock (RTC) circuit keeps the computer time/date up to date with the help of a battery and CMOS memory. This circuit maintains correct date and time even when the main AC supply to the computer is switched off.

Parallel ports:A parallel port is a type of interface found on computers for connecting peripherals. In computing, a parallel port is a parallel communication physical interface. It is also known as a printer port.

Serial ports:A serial port is a serial communication physical interface through which information transfers in or out one bit at a time. It isused for connecting external modems and older computer mouse.

USB ports: It can connect all kinds of external USB devices such as external hard disk, printer, scanner, mouse, keyboard etc.Most of the computers provide two USB ports as minimum.

Cables: There are mainly two types of cables

i) Data cables: A data cable is a cable that provides communication between devices.These cables are used to move data to and from the peripheral similar to the serial and parallel cables used to connect external devices to the computer. For example, the data cable that connects the monitor to computer allows to display a picture on the monitor. Other examples of data cables include the IDE, SATA, and USB cables.

ii) Power cables: The power supply cables provide the power needed by all the components of the computer to operate.

Floppy disk drive: Each floppy disk drive consists of a slot to accept a floppy disk, a master that spins the disk and a recording or reading device that moves across the disk in order to enter or read data.

Hard disk drive: Computer’s main permanent storage unit holding large amount of data and programs.

SMPS [switch mode power supply]: All components in a pc need electronic power supply. The SMPS converts the normal household electric AC supply to +5V and +12V DC supply. Usually +12V DC supply is connected to mother board, whereas +5V is connected to other peripheral devices such as floppy, hard disk, etc.

Expansion slots: An expansion slot is a socket on the motherboard that is used to insert an expansion card (or circuit board), which provides additional features to a computer such as video, sound, advanced graphics, Ethernet or memory. When you want to add a new device to your computer other than what is on the motherboard, you need an expansion slot. Depending on the width and the technology expansion slots can be divided into the following categories: 8 bit ISA, 16 bit ISA, PCI, AGP etc.

RESULT :

The hardware components of a computer are familiarized. A computer is basically an electronic device which converts unorganized facts on data into some meaningful information. Thus with the help of all these components the system performs well.

EXPERIMENT – 2

CALCULATOR OPERATIONS

OBJECTIVE:

To write a python program for performing various calculator operations

ALGORITHM:

```
Step 1:    Start
Step 2:    Input two numbers
Step 3:    Add two numbers
Step 4:    Subtract two numbers
Step 5:    Multiply two numbers
Step 6:    Divide two numbers
Step 7:    Print result of addition
Step 8:    Print result of addition
Step 9:    Print result of subtraction
Step 10:   Print result of multiplication
Step 11:   Print result of division
Step 12:   Stop.
```

PROGRAM:

```
a=input(“Enter First Number”)
b= input(“Enter Second Number”)
c=a+b
d=a-b
e=a*b
```

```
f=a/b
print “Addition of  Two Numbers is”,c
print “Subtraction of  Two Numbers is”,d
print “Multiplication of  Two Numbers is”,e
print “Division of Two Numbers is”,f
```

OUTPUT:

```
Enter first number number:10


    Enter Second number:5
   Addition of  Two Numbers is 15
   Subtraction of  Two Numbers is 5
   Multiplication of  Two Numbers is 50
  Division of Two Numbers is 2
```

RESULT :

The program is executed and the result obtained successfully.

EXPERIMENT – 3

ODD or EVEN

OBJECTIVE:

To write a python program to check whether given number is odd or even

ALGORITHM:

```
Step 1:    Start
Step 2:    Input a number
Step 3:    Check  if number%2=0 then
Step 4:     Print number is even
Step 5:     else
Step 6:    Print number is odd
Step 7:    Stop.
```

PROGRAM:

```
a=input("Enter one number:")
if a%2 is 0:
   print"Even"
else:
    print"Odd"
```

OUTPUT:

```
Enter one number:3
Odd
Enter one number:4
```

```
        Even
```

RESULT :

The program is executed and the result obtained successfully.

EXPERIMENT – 4

GREATEST AMONG THREE NUMBERS

OBJECTIVE:

To write a python program to find greatest amiong three numbers using nested if

ALGORITHM:

```
Step 1:   Start
Step 2:   Input three numbers
Step 3:   Check if a>b then check a>c
Step 4:   If step 3 is true then print a is greater
Step 5:   Else print c is greater
Step 6:   If a!>b then check b>c then print b is greater
Step 7:   Else print c is greater
Step 8:   Stop
```

PROGRAM:

```
a=input("enter data for a:")

b=input("enter data for b:")

c=input("enter data for c:")

if a>b:

   if a>c:

     print "a is greater"
```

```
  else:
    print "c is greater"
else:
    if b>c:
        print "b is greater"
    else:
        print "c is greater"
```

OUTPUT:

```
enter data for a:78
enter data for b:67
enter data for c:32
a is greater
```

RESULT :

The program is executed and the result obtained successfully.

EXPERIMENT – 5

PRIME or COMPOSITE

OBJECTIVE:

To write a python program to check whether given number is prime or not

ALGORITHM:

```
Step 9:    Start
Step 10:   Input a number
Step 11:   Set the flag as 0
Step 12:    Repeat the loop variable'i' in the range(2,number/2)
Step 13:   Check if number mod 'i' is 0 then
Step 14:    Set the flag as 1
Step 15:   Check if flag is 1 then
Step 16:   Print composite
Step 17:   Else print prime
Step 18:   Stop
```

PROGRAM:

```
n=input("enter anumber:")
flag=0
for i in range(2,n/2):
   if n%i is 0:
```

```
            flag=1
    if flag is 1:
       print "composite"
    elif flag is 0:
       print "prime"
```

OUTPUT:

```
          enter a number:11
         prime
        enter a number:12
        composite
```

RESULT :

The program is executed and the result obtained successfully.

EXPERIMENT – 6

FIBINOCCI SERIES

OBJECTIVE:

To write a python program to print the fibinocci series of ‘n’ numbers

ALGORITHM:

```
Step 1:    Input a number
Step 2:    Set the sum as 0,variable ‘a’ as 1,variable ‘b’ as 1
Step 3:    Set the loop variable ‘i’ as 2
Step 4:    Print the value in the variable sum
Step 5:     Check while  the loop variable i<n
Step 6:    If step 5 is true then
Step 7:    Set ‘a’ as ‘b’, ‘b’ as ‘s’, sum as ’a+b’
Step 8:    Print sum
Step 9:    Increment ‘i‘ by 1
Step 10:    Repeat step 5 to step p till i=n
Step 11:   Stop
```

PROGRAM:

```
n=input("enter a number")
a=1
b=1
s=0
i=2
print s,
while i<=n:
    a=b
```

```
        b=s
        s=a+b
        print s,
         i=i+1
```

OUTPUT:

```
    enter a number10

     0 1 1 2 3 5 8 13 21 34
```

RESULT :

The program is executed and the result obtained successfully.

EXPERIMENT – 7

FLOYD'S TRIANGLE

OBJECTIVE:

To write a python program to implement floyd's triangle

ALGORITHM:

```
Step 1:    Start
Step 2:    Enter the limit
Step 3:    Repeat the loop variable in the range(0,n)
Step 4:    Repeat loop variable in the range(0,i+1)
Step 5:    Print "*"
Step 6:    Repeat step 3 and 4 till i=n
Step 7:    Stop
```

PROGRAM:

```
n=input("enter limit")
for i in range(0,n):
    for j in range(0,i+1):
        print "*",
    print "\n"
```

OUTPUT:

```
enter limit5
  *

  * *

  * * *

  * * * *

  * * * * *
```

RESULT :

The program is executed and the result obtained successfully.

EXPERIMENT –8

HCF and LCM OF THE GIVEN NUMBERS

OBJECTIVE:

To write a python program to find the lcm and hcf of the given numbers

ALGORITHM:

```
Step 1: Start
Step 2: Read two numbers: a, b
Step 3: Assign the greatest number to “big” and smallest number to “small”.
Step 4: while “small” not equal to 0
Step 5: compute r = big % small
Step 6: small = big
Step 7: r = small
Step 8: end while
Step 9: Print “HCF=”  big
Step 10: LCM =(a*b)/ HCF
Step 10: Print LCM
Step 11: Stop
```

PROGRAM

```
a=input("Enter first number:")
b=input("Enter the second number:")
if(a>b):
        big=a
        small=b
else:
        big=b
        small=a
while(small!=0):
                r=big%small
                big=small
                small=r
print "Hcf is:",big
hcf=big
lcm=(a*b)/hcf
print "LCM is:",lcm
```

OUTPUT:

```
Enter first number:6
Enter the second number:3
Hcf is: 3
```

```
LCM is: 6
```

RESULT :

The program is executed and the result obtained successfully.

EXPERIMENT – 9

FACTORIAL OF A NUMBER

OBJECTIVE:

To write a python program to find factorial of a given number

ALGORITHM:

```
Step 1:   Start
Step 2:   Enter the number in ‘n’
Step 3:   Set fact variable as 1
Step 4:   Check while n>=1
Step 5:   If true find fact=fact*n
Step 6:   Decrement the value of n by 1
Step 7:   Repeat step 4 to step 6 till n=1
Step 8:   Print the value of fact
Step 9:   Stop
```

PROGRAM:

```
n=input("enter a number")
fact=1
while n>=1:
   fact=fact*n
   n=n-1
print "factorial of a given number is",fact
```

OUTPUT:

```
enter a number6
factorial of a given number is 720
```

RESULT :

The program is executed and the result obtained successfully.

EXPERIMENT – 10

STRING SLICING

OBJECTIVE:

To write a python program to perform different string slicing.

ALGORITHM:

```
Step 1:   Start
Step 2:   Input a string
Step 3:   print str[0:6]
Step 4:   print str[0:3]
Step 5:   print str[2:5]
Step 6:    print str[:6]
Step 7:   print str[3:]
Step 8:   Stop
```

PROGRAM:

```
str=raw_input("Enter a string")

 print str[0:6]

 print str[0:3]

 print str[2:5]

 print str[:6]

 print str[3:]
```

OUTPUT:

```
Enter a string nikhil
'Nikhil'

'Nik'

'khi'

'Nikhil'

'hil'
```

RESULT :

The program is executed and the result obtained successfully.

EXPERIMENT – 11

STRING OPERATIONS

OBJECTIVE:

To write a python program to perform different string operations.

ALGORITHM:

```
Step 1:   Start
Step 2:   Input a string
Step 3:   Input substring
Step 4:   Print count of substring in string
Step 5:   Display whether string ends with substring
Step 6:    Find substring in string
Step 7:   Find the index of string in substring
Step 8:   Check whether given string is all number
Step 9:   Check whether given string is all number
Step 10:  Check whether given string is all alphabet
Step 11:  Check whether given string is all in lowercase letters
Step 12:  Check whether given string is all uppercase letters
Step 13:  Check whether given string starts with substring
Step 14:  Display given string in swapcase form
Step 15:  Stop
```

PROGRAM:

```
msg = "welcome to sssit"

substr1 = "o"

print  msg.count(substr1)
print msg.endswith(substring1)

print msg.find(substr1)

print msg.index(substr1)

print msg.isalnum()

print msg.isalpha()

print msg.islower()

print msg.isupper()

print msg.lower()

print msg.upper()

print msg.startswith(substr1)

print msg.swapcase();
```

OUTPUT:

```
2

False

3

3

False
```

```
True

True

False

welcome to sssit

WELCOME TO SSIT

False

WELCOME TO SSIT
```

RESULT :

The program is executed and the result obtained successfully.

EXPERIMENT – 12

LIST OPERATIONS

OBJECTIVE:

To write a python program to perform various list operations.

ALGORITHM:

```
Step 1:     Start
Step 2:     Input lists in data1 and data2
Step 3:     Print first element of data1
Step 4:     Print first two  element of data1
Step 5:     Print first two  element of data2
Step 6:      Print elements of data1 using [0:]
Step 7:     Print two elements of data2[:2]
Step 8:     Print single replication of data1 using  data1*2
Step 9:     Append two list using data1.append(data2)
Step 10:    Display data1
Step 11:    Delete first element of data1
Step 12:    Display data1
Step 13:    Stop
```

PROGRAM:

```
data1=[1,2,3,4]

data2=['x','y','z']
```

```
print data1[0]
print data1[0:2]
print data2[-3:-1]
print data1[0:]
print data2[:2]
print data1*1
data1.append(data2)
print data1
del data1[0]
print data1
print “minimum value in list is”,min(data1)
print “maximum value in list is”,max(data1)
print "No. of elements in data1: ",len(data1)


//peplica of a list in another list//
z=[1,2,3]
c=[]
for i in z:
 c.append(i)
```

```
print “replica of list z in c is”,c
```

OUTPUT:

```
1
[1, 2]
['x', 'y']
[1, 2, 3, 4]
['x', 'y']
[1,2,3,4,1,2,3,4]
[1,2,3,4,['x','y','z'] ]
[1,2,3,4]
[2,3,4]
Minimum value of list is 1
maximum value in list is 4
No. of elements in data1 is 4
replica of list z in c is =[1,2,3]
```

RESULT :

The program is executed and the result obtained successfully.

EXPERIMENT – 13

DICTIONARY OPERATIONS

OBJECTIVE:

To write a python program to perform various dictionary operations and methods.

ALGORITHM:

Step 1: Start

Step 2 : Assign values to the dictionary, data. The values are assigned along with their identifier.

Step 3: Display the dictionary, data

Step 4: Update the dictionary. The entry corresponding to identifier 102 is updated as ‘ashok’

Step 5: Display the dictionary, data

Step 6: Access the value corresponding to identifier 100 from the dictionary, data

Step 7: Delete the entry corresponding to identifier 102 from the dictionary, data

Step 8: Print the dictionary, data

Step 9: Print the length of the dictionary, data

Step 10: Print the values of the identifier using data.keys()

Step 11: Print the values in the dictionary, data using data.values()

Step 12: Print the items of the dictionary, data using data.items()

Step 13: Copy the dictionary, data to a new dictionary, data1

Step 14: Display the new dictionary, data1

Step 15: Stop

PROGRAM:

```
Data={100:'Ram', 101:'Suraj', 102:'Alok'}
print data

//updation in dictionary//


data[102]='ashok'
print data

//accessing values from dictionary//


print "name of person having id 100 is",data[100]

//deletion//


del data[102]
print data
```

```
//dictionary methods

print len(data)
print data.keys()
print data.values()
print data.items()
data1=data.copy()
print data1
```

OUTPUT:

```
{100:'Ram', 101:'Suraj', 102:'Alok'}
{100:'Ram', 101:'Suraj', 102:'Ashok'}

ram

{100: 'Ram', 101: 'Suraj'}
2

[100, 101]
['Ram', 'Suraj']
[(100, 'Ram'), (101, 'Suraj')]
{100:'Ram', 101:'Suraj' }
```

RESULT :

The program is executed and the result obtained successfully.

EXPERIMENT – 14

IMPLEMENTATION OF FUNCTION

OBJECTIVE:

To write a python program to perform addition and subtraction of two numbers using single function

ALGORITHM:

```
Step 1:    Start
Step 2:    Input two numbers
Step 3:    Call add function
Step 4:    Addition and subtraction of two numbers is performed inside the
    function
Step 5:    Function return added and subtracted value to the called function
Step 6:    Display the result
Step 7:     Stop
```

PROGRAM:

```
def add(x,y):

        return a+b,a-b

a,b=input("enter the numbers")

print "sum and difference  is",add(a,b)
```

OUTPUT:

```
Enter two numbers 5 4

9   1
```

RESULT :

The program is executed and the result obtained successfully.

EXPERIMENT – 15

FILE OPERATIONS

OBJECTIVE:

To write a python program to perform various file operations.

ALGORITHM:

```
Step 1:    Start
Step 2:    open the file
Step 3:    print the contents of the file
Step 4:    write a text into the file using f.write
Step 5:    close
Step 6:    open the file in read mode
Step 7:    read the content of the file into the variable
Step 8:    print the content of a file using that variable
```

PROGRAM:

```
f = open("test.dat","w")

print f

f.write("Now is the time")

f.write("to close the file")

f.close()

f = open("test.dat","r")

text=f.read(5)
```

```
print text
```

OUTPUT:

```
<open file 'test.dat', mode 'w' at 0x7f5a996fa540>
Now i
```

RESULT :

The program is executed and the result obtained successfully.