Experiment no: 01

Experiment name: To write a program in "C" or "Java" or "Python", to develop a simple calculator that would be able to take a number, an operator (addition / subtraction / multiplication / division / modulo) and another number consecutively as input and the program will display the output after pressing "=" sign.

Algorithm:

step-1: Start

step-2: Enter an operator such as (+, -, *, /, %)

step-3: Enter operands

step-4: Use switch(){ case: ... } operation.

step-5: Check each case & Until get appropriate match. if no match is occoursed then print defaut output.

step-6: Save and Exit.

Source Code:

```
#include <stdio.h>
int main(){
        char op;
        double first, second;
        printf("Enter an operator (+, -, *, /, %): ");
        scanf("%c", &op);
        printf("Enter two operands: ");
        scanf("%lf %lf", &first, &second);
    switch(op){
        case '+':
            printf("%.1lf + %.1lf = %.1lf", first, second, first + second);
            break;
        case '-':
            printf("%.1lf - %.1lf = %.1lf", first, second, first - second);
            break;
        case '*':
            printf("%.1lf * %.1lf = %.1lf", first, second, first * second);
            break;
        case '/':
            printf("%.1lf / %.1lf = %.1lf", first, second, first/second);
            break;
        case '%':
            printf("%.1lf % .1lf = %.1lf", first, second, first % second);
            break;
        default:
            printf("Error! Operator is not correct");
    }
    return 0;
}
```

Source Code:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
void numberGenerate(char lastDigit, int *num){
        *num = *num*10 + (int)lastDigit-48;
}
float calculate(int a, int b, char s){
    switch(){
            case '+':
                return a+b;
            case '-':
                return a-b;
            case '*':
                return a*b;
            case '/':
                return a/b;
            case '%':
                return a%b;
            }
    }
int main(){
    char *data;
    int num1=0, num2=0, flag=0;
    char d, sign;
    for(d=getch(); d!='='; d=getch()){
        printf("%c", d);
        if(d=='+'||d=='-'||d=='*'||d=='/'||d==
```

```
    {flag = 1;
     sign = d;
     continue;
    }
if (flag == 0){
   numberGenerate(d, &num1);
else {
   numberGenerate(d, &num2);
   }
}
printf("\n%d %c %d = %.2f", num1, sign, num2, calculate(
                    num1, num2, sign));

return 0;
}
```

Experiment no: 02

Experiment name: To write a c program that will take two 'n' integers as input until a particular operator and produce 'n' output.

Theory: Algorithm:

Step-1: Start

Step-2: Take user input from user and assigne data into each variable.

Step-3: Use if condition calculate data length and again use if condition for calculating particular operator (+, -, *, /, %) performance

Step-4: If_elif condition use for produce 'n' output.

Step-5: Save and exit.

Source code:

```
inp = input("Enter data: ");
data = inp.strip()
data = data.split()
sign = data.pop()
output = []
if len(data)%2 == 0:
    if sign == '+':
        for i in range(0, len(data), 2):
            output.append(int(data[i])+int(data[i+1]))
    elif sign == '-':
        for i in range(0,2, len(data)):
            output.append(int(data[i])-int(data[i+1]))
    elif sign == '*':
        for i in range(0,len(data),2):
            output.append(int(data[i])*int(data[i+1]))
    elif sign == '/':
        for i in range(0, len(data),2):
            output.append(int(data[i]/int(data[i+1]))
    print(output)
else:
    print("Please enter pair number")
```

Experiment no: 03

Experiment name: To write a program in 'C' or 'Java', ~~Python~~ to check whether a number or string is palindrome or not.

Theory:

Palindrome number: A palindrome number is a number that remains the same when its digits are reversed. So, It has reflectional symmetry across a vertical axis. The term palindromia is derived from palindrome, which refers to a word whose spelling is unchanged when it's letters are reversed.

Algorithm:

step-1: Start.

step-2: Declear variable and store data into it.

step-3: Using while loop for produce remainder and reversed number.

step-4: Use if for checking reversed number is equal or not equal to original number.

step-4: If it is equal, then Print Palindrom else, print, not a palindrom.

step-5: Save and exit.

Source Code:

```
#include <stdio.h>
int main(){
    int n, reversed = 0, remainder, original;
    printf("Enter an integer: ");
    scanf("%d", &n);
    original = n;
// reversed integer is stored in reversed variable
    while(n!=0){
        remainder = n%10;
        reversed = reversed*10 + remainder;
        n /= 10;
    }
// palindrome if original and reversed are equal
    if(original == reversed)
        printf("%d is a palindrome.", original);
    else printf("%d is not palindrome.", original);

    return 0;
}
```

Experiment name: To write down the ATM system specifications and report the various bugs.

Solution: Automated Teller Machine (ATM): An automated teller machine (ATM) is an electronic banking out let that allows customers to complete basic transactions without the aid of a branch representative or teller. Anyone with a credit card or debit card can access cash at most ATMs. ATM's are convenient allowing customers to perform quick self service transactions such as deposits, cash withdrawls, bill payments and transfer between accounts.

System Specifications of ATM:

i) When the machine is idle, a greeting message is displayed.

ii) The keys and deposit slot will remain inactive untill a bank card has been entered.

iii) When a bank card is inserted, the card reader attempts to read it.

iv) If the card can not be read, the user is informed that the card is unreadable.

v) Then the card is ejected.

vi) If the card is readable, the card reader reads the account and PIN numbers off the card and the user asked to enter his or her PIN.

vii) The user can select a transaction among deposit funds, withdraw funds, transfer funds, Query the balance of any account and specify all relevant information.

viii) When a transaction has been completed, the system returns to the main menu.

The various bugs of the ATM system is given below:

i) faulty Dispenser. A rarce, but exceedingly frustrating issue that can occur is an ATM that has a faulty dispenser.

ii) Worn out Card Reader. Every bank card or credit card has a dark stripe on the back.

iii) Broken keypad.

iv) Receipt Malfunctions.

v) Software Glitches.

Experiment no: 05

Experiment name: To write a "c" program to find out the factorial of a number using while and for loop. Also verify the results obtained from each case.

Theory:

Factorial number: The factorial number of a given number is the product of all a given number is the product of all the integers from 1 to that numbers.

$$6! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 = 720$$

factorial is not defined for negative numbers, and the (!) factorial of zero(0!) is one(1). $0! = 1$.

Algorithm:

Step-1: Start

Step-2: Declear variable and take input from user.

Step-3: Use for loop for calculating factorial,

```
for (i=1; i<=Number; i++){
    factorial = factorial * i;
}
```

Step-4: Display output. Again, Use while loop, while(i<= numbers){factorial = factorial * i; i++;}

Step-5: Save and exit.

Source Code:

```
#include<stdio.h>      /* Using for loop. */
int main(){
    int i, Number;
    long Factorial=1;
    printf("\n Please Enter any number to find Factorial\n");
    scanf("%d", &Number);
    for(i=1; i<=Number; i++){
        Factorial = Factorial * i;
    }
    printf("\nFactorial of %d = %d\n", Number, Factorial);
    return 0;
}
```

```
/* factorial. using While loop. */
# include <stdio.h>
int main(){
    int Number, i=1;
    long Factorial=1;
    printf("\n Please Enter any number to find Factorial\n");
    scanf("%d", &Number);
    while(i<=Number){
        Factorial = Factorial * i;
        i++; }
    printf("Factorial of %d = %d\n", Number, Factorial);
    return 0;
}
```

Experiment no: 06

Experiment name: To write a 'c' program that will find sum and average of an array using do while loop and 2 user defined function.

Objectives: To find sum and average of an array.

Algorithm :

step-1 : start

step-2 : Declear variable such as, n, number, Sum float Average;

step-3 : Use do {...} while( ); loop for calculation.

step-4 : Display Sum and Average

step-5 : Save and exit.

Experiment no: 07

Experiment name: To write a simple "Java" program to explain classNotfound Exception and endOffile (EOF) exception.

Algorithm:

step-1: start-

step-2: Define public class exp7
Define " " exp7_2
Define " " abc

step-3: Using try{...} catch{--} handelled exception of error.

step-4: Using try{--} catch{...} handelled exception of error, for, classNotfound, IOException, EOFException.

step-5: Display output.

step-6: Save and exit.

```
public static void main (String[ ] args) throws Exception{
    System.out.println ("Enter a String: ");
    Scanner sc = new Scanner (System.in);
    String data = sc.nextLine( );
    byte[ ] buf = data.getBytes ( );
    DataOutputStrem dos = new DataOutputStream (new
    fileOutputStream ("doc.txt");
    for(byte b : buf){
        dos.writeChar (b);
    }
    dos.flush( );
DataInputStream dis = new DataInputStream(new fileInputStream
        ("abc.txt"));

while(true){
    char ch;
    try { ch = dis.readChar( );
        System.out.print (ch);
    }
    catch (EOFException e) {
        system.out.println (" ");
        system.out.println ("End of file reached");
        break;
    }
    catch (IOException e){
        system.out.println (e);
    }
}
}
```

Source Code:

```
#include <stdio.h>
int main()
{ int n, numbers, i=0, Sum = 0;
  float Average;
  printf ("\n Please Enter How many Numbers you want?\n");
  scanf ("%d", &n);
  printf ("\n Please Enter the elements one by one\n");
do { scanf ("%d", &numbers);
     Sum = Sum + numbers;
     i++;
   }
while (i<n);
Average = Sum/n;
printf ("\n Sum of the %d Numbers = %d ", n, Sum);
printf ("\n Average of the %d Numbers = %.2f", n, Average);
return 0;
}
```

Source Code :

```
Package softwareTesting;
public class exp7{
        public static void main(String[ ] args)
        { try {
            Class.forName("dbc");
          }
          catch (Exception e){
              System.out.println(e);
          }
      }
  }
```

Again,

```
Package softwareTesting;
public class abc{
    public static void main (String[ ] args){
        System.out.println ("abc is available");
    }
}
```

And Also, Again,

```
package softwareTesting;
import java.io.*;
import java.util.*;
public class exp7_2{
```

Experiment no: 08

Experiment name: To write a "C" or "Java", "Python" program that will read a input.txt file containing n positive integers and calculate addition, subtraction, multiplication and division in separate output.txt file.

Algorithm:

step-1: Start

step-2: open input.text(inp.text) file and assign in f.

step-3: Now, read input text file and assign it into another variable.

step-4: print read file.

step-5: Open output.text

step-6: Use if condition and for loop for performing calculation.

step-7: Display output and close file by using close()

step-8: Save and exit.

Source Code :

```
f = open ("inp.text", "r")
file_read = f.readline()
f.close()
print(file_read)
inp = str(file_read)
print(inp)
data = inp.strip()
data = data.split()
f = open ("output.txt", "w"
if len(data)%2 == 0:
    l = 0
    for i in range(0, len(data), 2):
        l = l+1
        output = [ ]
        output.
        output.
        output.
        output.
        ans = ' '.join(output)
        line = "case " + str(l) + ': ' + ans + '\n'
        f.write(line)
        print(output)
    f.close()
else:
    print ("Please enter pair number")
```

Question no: 01 (Q-01)

Question name: To explain the role of software engineering in Biomedical engineering and in the field of Artificial Intelligence and Robotics.

Solution:

The Role of software engineering: According to the IEEE Engineering in Medicine and Bidogy Society (EMBS), engineering in biomedicine is a fast growing specifity. Software engineers are important facts of biomedical engineering and science. Most medical devices are required software to function. Developing and maintaining that software is an important job of the biomedical software engineer. Biomedical researchers are look to software engineer to develop algorithms for data analysis and biological system modeling. There is a great need for software engineers in the field of image and signal processing for biomedicine. Software engineers are involved in the collection and analysis of biomedical information collected by clinicans and researchers.

Software engineering plays a vital role in the field of Artificial Intelligence and Robotics. The need of software engineering in this field are growing increasingly in

demand. Software engineers design, create, test and manage software systems for artificial intelligence programs or applications using a variety of programming languages Software engineeringe don't only write code, they also design everything from the ground up. software engineers also collaborate with other IT professionals that the system meet specific requirements.

Question no: 02 (Q-02)

Question name: To study the various phase of water-fall model. which phase is the most dominated one?

Solution: Water-fall Model: Waterfall model is the simplest model of software development paradigm. All the phases of SDLC will function one after another in linear manner. That is when the first phase is finished then only the second phase will start and so on.

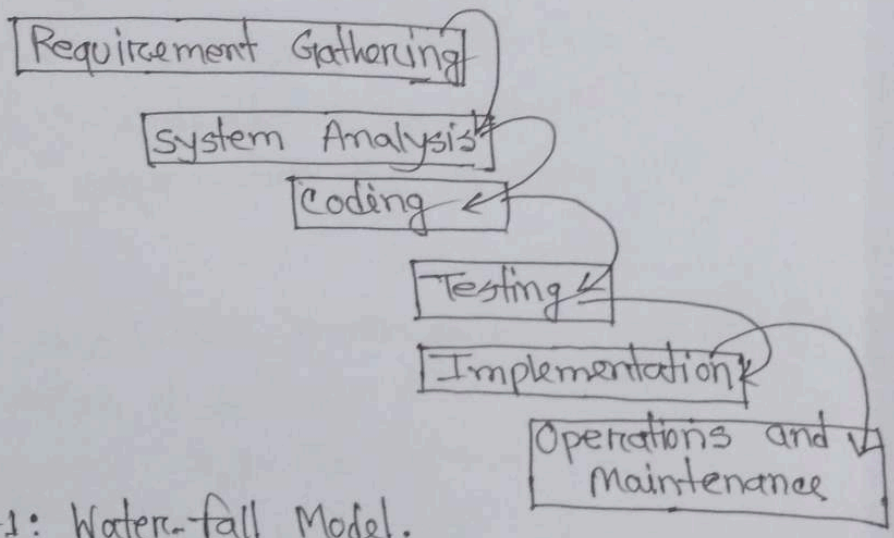

fig-1: Water-fall Model.

The study of the various phase of waterfall model is written below:

Requirement Gathering: This step onwards the software development team works to carry on the project.

The team holds discussions with various stake holders from problem domain and tries to make bring out as much information as possible on their requirements. The requirements are contemplated and segrated into user requirements, system requirements, and functional requirements.

System Analysis: At this step, the developers decide a roadmap of their plan and try to bring up the software model suitable for the project. System analysis includes understanding of software product limitations, learning system related problems or changes to be done in existing system beforehand, identifying and addressing the impact of project on organization and personnel etc.

Coding: This step is also known as programming phase. The implementation of software design starts in terms of writing program code in the suitable programming language and developing error-free executable programs efficiently.

Testing: An estimate says that 50% of whole software development process should be tested. Errors may ruin the software from critical level of its own removal.

software testing is done while coding by the developers and through testing is conducted by testing experts at various levels of code such as module testing, program testing, product testing, testing the product at user's end etc.

Implementation: This means installing the software on user machine. At time software needs post-installation configurations at user end. Software is tested for portability and adaptability and integration related issues are solved during implementation.

Operations and Maintenance: This phase confirms the software operation in terms of more efficiency and less errors. If required the user's are trained on. The software is maintained timely by updating the code according to the changes taking place in user end environment or technology.

In my opinion, the system analysis phase is the most dominated one. Because in this phase, the team decide a roadmap of the full system start to end. So every, possible risk, errors, advantages, disadvantages, software model will be discuss here. That's why it's the most dominated one, in my opinion.

Question no: 03 (Q-03)

Question name: Using COCOMO model estimate effort for specific problem in industrial domain.

Solution: CoCoMo Model: CoCoMo (Constructive Cost Model) is a regression model based on LOC (Number of line of code). It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. It was proposed by Barry Boehm in 1970.

Using the basic CoCoMo model, the estimation of effort calculations are given below:

$$\text{Person required} = \frac{\text{Effort}}{\text{time}} ; \quad E = a(KLOC)^b ; \quad \text{time} = c(\text{Effort})^d$$

The above formula is used for the effort estimation of the basic CoCoMo model. The ~~eos~~ constant values a, b, c for the basic model for the different categories of the system is fixed but different.

The effort is measured in person-months and as the evident from the formula is dependent on kilo-lines of code. The development time is measured in months.

Using these formula, we can estimate effort for a specific problem in industrial domain.

Question no: 01 (Q-01)

Question name: To identify the reasons behind software crisis and explain the possible solutions for the following scenario:

case-01: Air ticket reservation software was delivered to the customer and was installed in an airport at 12.00 AM (mid night) as per the plan. The system worked quite fine till the next day 12.00 PM (noon). The system crashed at 12.00 and the airport authorities could not continue using software for ticket reservation till 05.00 PM. It took five hours to fix the defect in the software.

case-02: Software for financial system was delivered to the customer. Customer conformed the development team about a mal-function in the system. As the software was huge and complex, the development team could not identify the defect in the software.