

## Index

01. Write a Program to develop a simple calculator that would be able to take a number, an operator and another number consecutively as input and the program will display the output after pressing “=” sign.
02. Write a program that will take two ‘n’ integers as input until a particular operator and produce ‘n’ output.
03. Write a program to check whether a *number* or *string* is palindrome or not.
04. Write down the ATM system specifications and report the various bugs.
05. Write a program to find out the factorial of a number using while and for loop. Also verify the results obtained from each case.
06. Write a “C” program that will find sum and average of an array using do while loop and 2 user defined function.
07. Write a simple program to explain classNotFound Exception and endOfFile(EOF) Exception.
08. Write a program that will read a *input.txt* file containing *n* positive integers and calculate addition, subtraction, multiplication and division in separate *output*.
09. Explain the role of software engineering in Biomedical Engineering and in the field of Artificial Intelligence and Robotics.
10. Study the various phases of Water-fall model. Which phase is the most dominated one?
11. Using COCOMO model estimate effort for specific problem in industrial domain.
12. Identify the reasons behind software crisis and explain the possible solutions for the following scenario:  
Case-1: “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 PM 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-2: “Software for financial systems 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.”

## 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 numbers, an operator (addition / subtraction / multiplication / division) and another number consecutively as input pairing "=" 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 occurred then print default output .

Step-6: Save and Exit

## Source code

```
def add(x, y):
    return x + y
def subtract(x, y):
    return x - y
def multiply(x, y):
    return x * y
def divide(x, y):
    return x / y
def modulo(x, y):
    return x % y

print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")
print("5.modulo")
while True:
    # take input from the user
    choice = input("Enter choice(1/2/3/4/5): ")

    # check if choice is one of the four options
    if choice in ('1', '2', '3', '4', '5'):
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))

        if choice == '1':
            print(num1, "+", num2, "=", add(num1, num2))

        elif choice == '2':
            print(num1, "-", num2, "=", subtract(num1, num2))

        elif choice == '3':
            print(num1, "*", num2, "=", multiply(num1, num2))

        elif choice == '4':
            print(num1, "/", num2, "=", divide(num1, num2))
        elif choice == '5':
            print(num1, "%", num2, "=", modulo(num1, num2))
    else:
        print("Invalid Input")
```

## Input and output

Select operation.

1.Add

2.Subtract

3.Multiply

4.Divide

5.modulo

Enter choice(1/2/3/4/5): 1

Enter first number: 5

Enter second number: 15

$5.0 + 15.0 = 20.0$

Enter choice(1/2/3/4/5):

## Experiment no. 02

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

Algorithm:

Step-1: Start

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

Step-3: Use if condition calculate data length and assign and again use if condition for calculate particular operator performance.

Step-4: If-elif condition 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")
```

## Input and output

Please enter pair number:

4 5 7 8

10 15

## 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 numbers: In a number that remains the same when its digits are reversed. So it has reflective symmetry across a vertical axis. The term palindrome is derived from palindrome which refers to a word whose spelling is unchanged when its letters are reversed.

Algorithm:

Step-1: Start

Step-2: Declares variable and store data.

Step-3: Using while loop for produce remainders and reversed number

Step-4: Use if for checking reversed numbers is equal or not equal to original number

Step-5: If it is equal, then print palindrome else print not a palindrome.

Step-6: Save and exit.

## Source code

```
data = input("Enter data: ")

if data == data[::-1]:
    print("It is palindrome")
else:
    print("It is not palindrome")
```

## Input and output

Enter data: 121

It is palindrome

Enter data: abc

It is not palindrome

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 electric banking outlet that allows customers to complete on 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. ATMs are convenient allowing customers to perform quick self-service transactions such as deposits, cash withdrawls, bill payment and transfers 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 until 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't 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 number off the card and the user is asked to enter his 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 rare but exceedingly frustrating issue that can occur in an ATM that has a faulty dispenser.
- ii) Worn out Card Reader, Every bank card on 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 numbers: The factorial number of a given number is the product of all a given number is the product of all integers from 1 to that number.

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

## Algorithm:

Step-1: Start

Step-2: Declare 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<=number)  
{ factorial = factorial \* i; i++ }

Step-5: Save and exit.

## Source code

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

```
fact = 1;
for i in range(1,num+1):
    fact = fact*i
print(fact)
```

```
fact = 1;
i = 1
while i <= num:
    fact = fact*i
    i = i + 1
```

## Input and output

Enter a number: 6

720

Enter a number: 8

40320

## 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: Declares 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 .

## Source code

```
#include<stdio.h>

int sumation(int data[], int l)

{
    int s = 0;
    for(int i=0; i<l; i++)
        s = s + data[i];
    return s;
}

float average(int s, int l)

{
    return s/l;
}

int main()

{
    int num[100], i, len, sum;
    float avg;
    printf("How many number you want to input: ");
    scanf("%d", &len);
    for (i = 0; i<len; i++)
        scanf("%d", &num[i]);
    i = 0;
    sum = 0;
    do{
        sum = sum + num[i];
        i++;
    }
```

```
while(i<len)
    avg = sum/len

    printf("Sum = %d\n", sum);
    printf("Average = %f\n", avg);
```

```
sum = summation(num, len);
avg = average(sum, len);
```

```
printf("Function: Sum = %d\n", sum);
printf("Function: Average = %f\n", avg);
```

```
}
```

## Input and output

How many number you want to input;

2,5,6,8

Sum=21

Average=5.25

Experiment name: no. 07

Experiment name: To write a simple "Java" program to explain class Not found exception and end of file (EOF) exception.

Algorithm:

Step-1: Start

Step-2: Define public class exp7.

Define " " exp7-2

Define " " abc

Step-3: Using try{....} catch{...}  
handled exception of errors.

Step-4: Using try{....} catch{...}  
handled exception of errors,  
for, class Not found, IOexception,  
EOFException,

Step-5: Display output.

Step-6: Save and exit.

## Source code

```
public class Example {  
    public static void main(String args[]) {  
        try {  
            Class.forName("NoClassExist");  
        } catch (ClassNotFoundException ex) {  
            ex.printStackTrace();  
        }  
    }  
}
```

## Input and output

java.lang.ClassNotFoundException: NoClassExist

## Experiment no-8

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 and generate output.txt file.

### Algorithm:

Step-1: Start

Step-2: Open input.txt (inp.txt) file and assign it to f.

Step-3: Now, read input.txt file and assign it to another variable.

Step-4: Print read file

Step-5: Open output.txt.

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.txt", "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.append(str(int(data[i])+int(data[i+1])))
        output.append(str(int(data[i])-int(data[i+1])))
        output.append(str(int(data[i])*int(data[i+1])))
        output.append(str(int(data[i])/int(data[i+1])))
        ans = ''.join(output)
        line = "case " + str(l) + ':' + ans + '\n'
        f.write(line)
        print(output)

f.close()
else:
    print("Please enter pair number")
```

## Input and output

10 5

10 5

['15', '5', '50', '2.0']

output.txt - Notepad

File Edit Format View Help  
case 1: 15 5 50 2.0

inp.txt - Notepad

File Edit Format View Help

10 5

Question no. 01 (Q-01)

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

Solution:

The Role of software engineering.

According to the IEEE Engineering in medicine and Biology society, engineering in biomedicine is a fast growing speciality. Software engineering are important facts of biomedical engineering and science. Most medical devices, are required software function. Developing and maintaining that software to function. Developing and maintaining that software in an important job of the biomedical software engineer. Biomedical researchers are look to software engineers to develop algorithm for data analysis.

and Biological systems modeling. 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 and manage software systems for artificial intelligence program. Software engineers 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.

and Biological system modeling. 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 and manage software systems for artificial intelligence program. Software engineers 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 that is when the first phase is finished then only the second phase will start and so on.

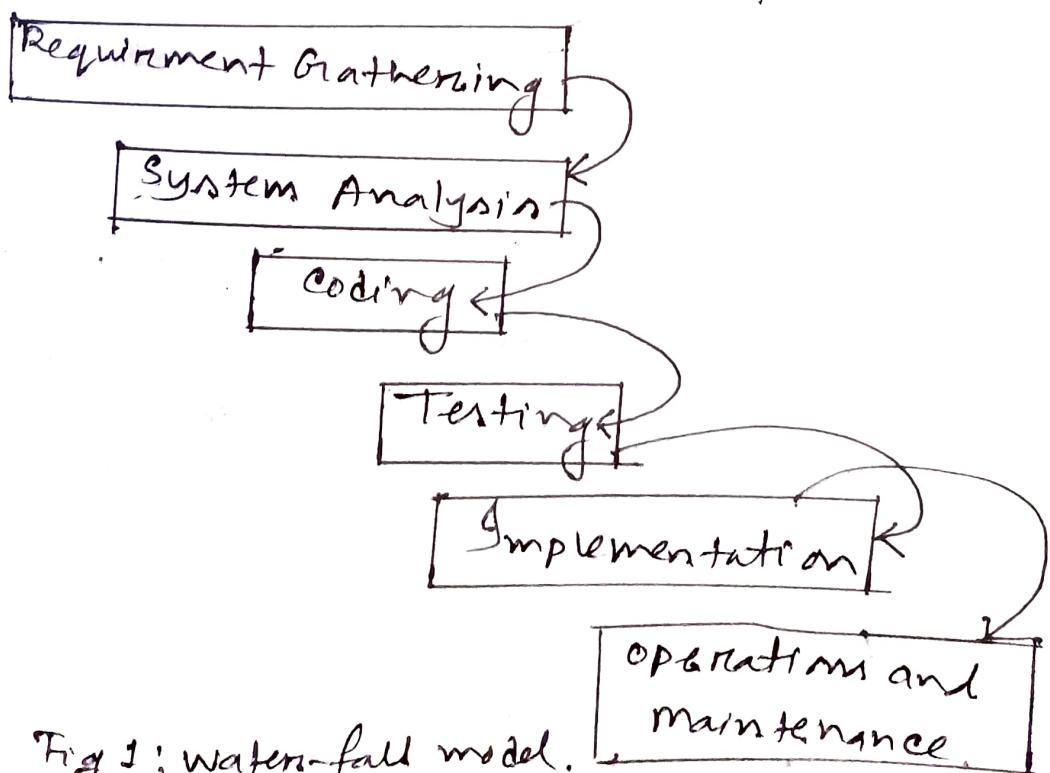


Fig 1: water-fall model.

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 requirements are contemplated and segregated into user requirements, system requirements and functional requirements.

System Analysis: At this stage, the developers decide a road map of their plan and try to bring up the software model suitable for the project.

Coding: This step is also known as programming phase. The implementation of software design starts in terms of writing code in software error-free, language and make it

testing: Software testing is done while coding by the developers and thorough testing is conducted by testing experts such as module testing, program testing.

Implementation: This means installing the software on user machine. 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.

In my opinion, the system analysis phase is the most dominated because in this phase, the team decide a road map of the full system start to end. So every possible risk will discuss here. That's why it's the most dominated 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 is a regression model based on LOC. 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.

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

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

The above formula is used for the effort estimation of the basic COCOMO model. The 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 is evident from the formula it depends on kilo-lines of code. The development time is measured in months. Using these formulas we can estimate effort for a specific problem in industrial domain.

## Question no-04 (Q-04)

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

Case-01: Air ticket reservation software was delivered tonight as per the plan. The system worked quite fine till the next day. 12:00 PM (Noon). The system crashed at 12:00 PM and the airport authorities could not continue using software for ticket reservation till 5:00 pm. It took five hours to fix the defect in the software.

Case-02: Software for financial system was delivered to the customers. Customers confirmed the development team about a malfunction in the system. As the software was huge and complex, the development team couldn't identify the defect in the software.

Background theory: In the early years of computer applications the focus of the development and innovation were on hardware, software was largely viewed as an after thought computer programming was an art. Programmers didn't follow any disciplined or formalized approaches. This way of doing things was adequate for a while until the sophisticated of computer applications outgrew. Software development projects produced thousand of source program statements with the increase in the size and complexity of the software. Following situations resulted in collectively termed as software crisis.

① Time Slippage

② Cont Slippage

③ Failure customers site

④ Intractable Errors after delivery.

Case-1 Solutions: In the crisis of the software on air ticket reservation "failure at customers site." In this situation the customers can't do their checking or other necessary work on this air station for failure the reservation software.

It took five hours to fix, so five hours the customers site failure.

case 2 solution: we can see that the case-2 scenario a user con formed product by the development team about mal-function in the system. As the software become huge complex. This is called the "Interactive Errors after delivery".

After delivery product some problem can be arises, which are unknown to the development team or the development team think that it is usual event but customer face this problem very much.