PRINCIPLES OF PROGRAMMING LANGUAGES

Practical File

ETCS-458

Submitted To
Dr. Vinita Rohilla
(Assistant Professor, CSE)

Submitted By
Name –**Bhoomaahamso Dharamalayenam**
Serial No. – **36**
Enrolment No. **- 14315002719**
Branch – **CSE-III**
Semester – **8th Semester**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY
(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI)
DELHI-110058

2019-2023

INDEX

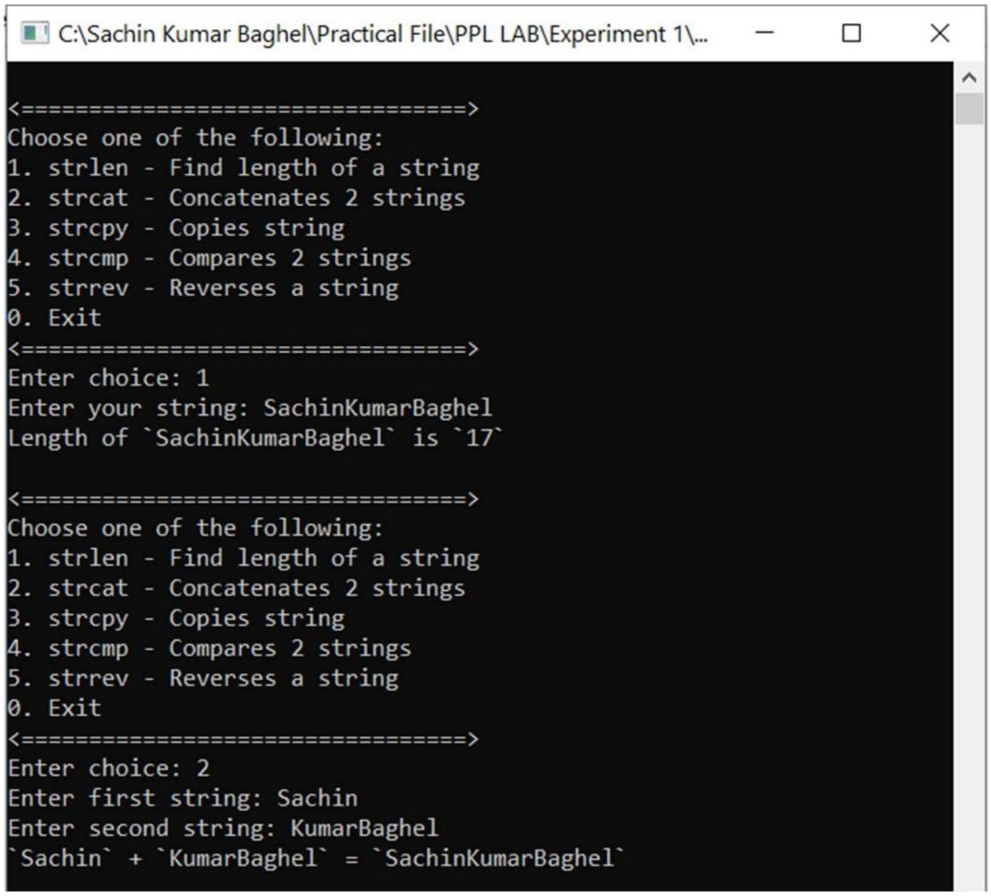
Experiment – 1

Aim - Implement all major functions of string.h in single C program using switch case to select specific function from user choice (strlen, strcpy,strcat, strncat,strcmp);

Program

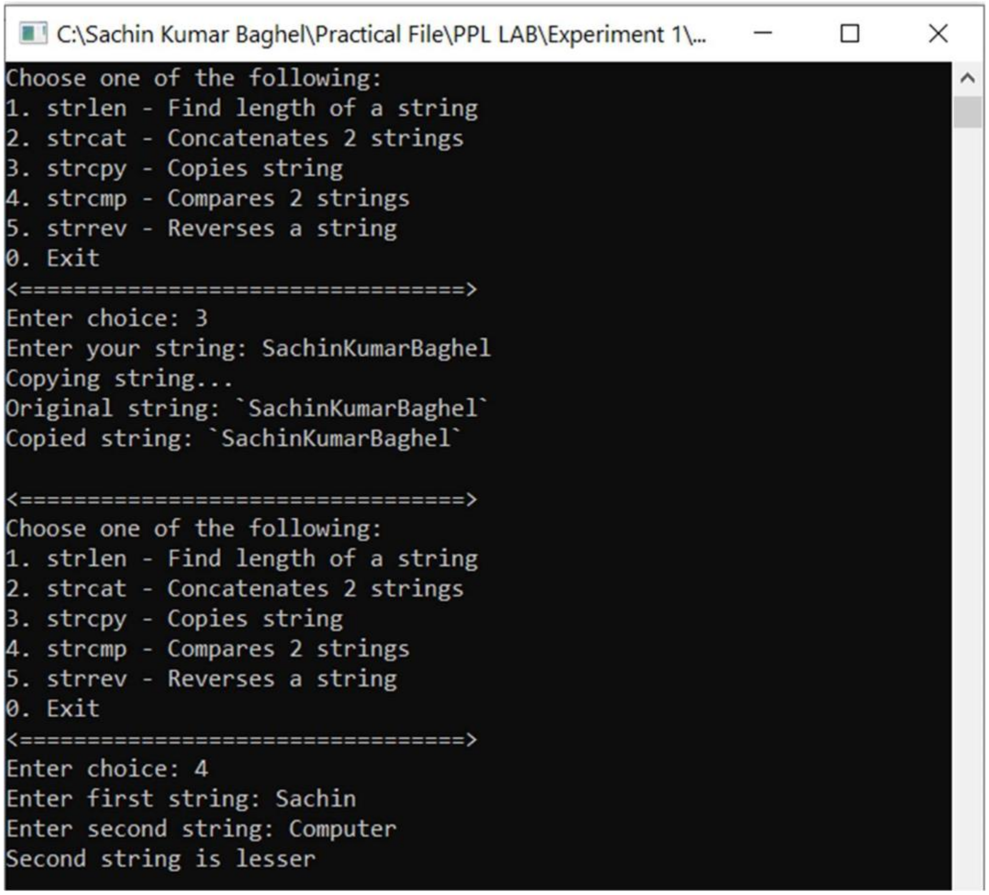
```
#include <stdio.h>
#include <string.h>
#define MAX_SIZE 50
int printMenuAndGetChoice()
{
    printf("\n<==================================>\n");
    printf("Choose one of the following:\n");
    printf("1. strlen - Find length of a string\n");
    printf("2. strcat - Concatenates 2 strings\n");
    printf("3. strcpy - Copies string\n");
    printf("4. strcmp - Compares 2 strings\n");
    printf("5. strrev - Reverses a string\n");
    printf("0. Exit\n");
    printf("<===============================>\n");
    int  choice;
    printf("Enter choice: ");
    scanf("%d", &choice);
    return choice;
}
void stringLength()
{
    char str[MAX_SIZE];
    printf("Enter your string: ");
    scanf("%s", str);
    printf("Length of `%s` is `%d`\n", str, strlen(str));
}
void concatenateStrings()
{
    char str1[MAX_SIZE], str2[MAX_SIZE];
    printf("Enter first string: ");
    scanf("%s", str1);
    printf("Enter second string: ");
    scanf("%s", str2);

    printf("`%s` + `%s` = ", str1, str2);
    printf("`%s`\n", strcat(str1, str2));
}
void copyStrings()
{
    char str1[MAX_SIZE], str2[MAX_SIZE];
    printf("Enter your string: ");
    scanf("%s", str1);
    printf("Copying string...\n");
    strcpy(str2, str1);
    printf("Original string: `%s`\n", str1);
    printf("Copied string: `%s`\n", str2);
}
void compareStrings()
{
    char str1[MAX_SIZE], str2[MAX_SIZE];
    printf("Enter first string: ");
    scanf("%s", str1);
    printf("Enter second string: ");
    scanf("%s", str2);
    int cmp = strcmp(str1, str2);
    if (cmp < 0)
    {
        printf("First string is lesser\n");
    }
    else if (cmp > 0)
    {
        printf("Second string is lesser\n");
    }
    else
    {
        printf("Both the strings are same\n");
    }
}
void reverseString()
{
    char str[MAX_SIZE];
    printf("Enter your string: ");
    scanf("%s",  str);
    strrev(str);
    printf("Reversed string: `%s`\n", str);
}
int main()
{
    int choice = -1;
    while (choice != 0)
    {
        choice = printMenuAndGetChoice();
        switch (choice)
        {
        case 1:
            stringLength();
            break;
        case 2:
            concatenateStrings();
            break;
        case 3:
            copyStrings();
            break;
        case 4:
            compareStrings();
            break;
        case 5:
            reverseString();
            break;
        case 0:
            break;
        default:

            printf("Enter correct choice\n");
        }
    }
    return 0;
}
```

OUTPUT

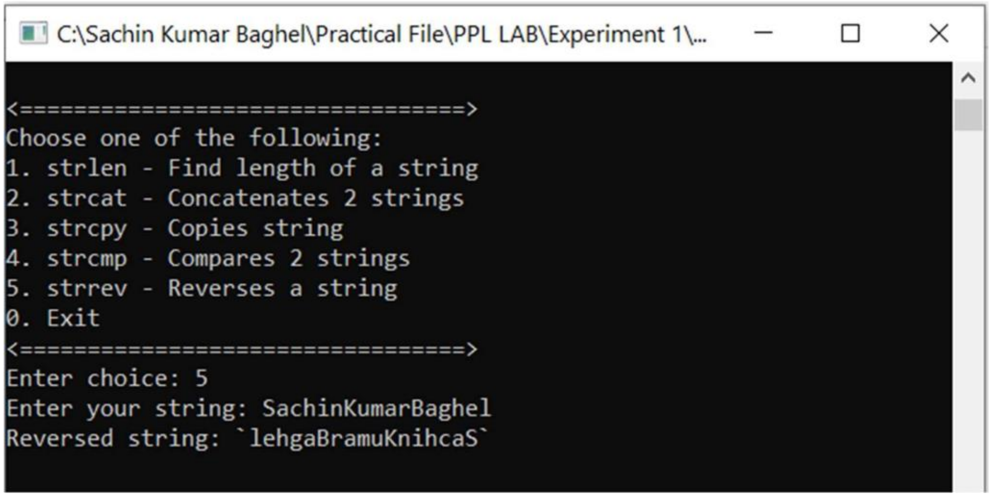


C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 1\...

```
<==================================>
Choose one of the following:
1. strlen - Find length of a string
2. strcat - Concatenates 2 strings
3. strcpy - Copies string
4. strcmp - Compares 2 strings
5. strrev - Reverses a string
0. Exit
<==================================>
Enter choice: 1
Enter your string: SachinKumarBaghel
Length of `SachinKumarBaghel` is `17`

<==================================>
Choose one of the following:
1. strlen - Find length of a string
2. strcat - Concatenates 2 strings
3. strcpy - Copies string
4. strcmp - Compares 2 strings
5. strrev - Reverses a string
0. Exit
<==================================>
Enter choice: 2
Enter first string: Sachin
Enter second string: KumarBaghel
`Sachin` + `KumarBaghel` = `SachinKumarBaghel`
```


C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 1\...

```
Choose one of the following:
1. strlen - Find length of a string
2. strcat - Concatenates 2 strings
3. strcpy - Copies string
4. strcmp - Compares 2 strings
5. strrev - Reverses a string
0. Exit
<=================================>
Enter choice: 3
Enter your string: SachinKumarBaghel
Copying string...
Original string: `SachinKumarBaghel`
Copied string: `SachinKumarBaghel`

<=================================>
Choose one of the following:
1. strlen - Find length of a string
2. strcat - Concatenates 2 strings
3. strcpy - Copies string
4. strcmp - Compares 2 strings
5. strrev - Reverses a string
0. Exit
<=================================>
Enter choice: 4
Enter first string: Sachin
Enter second string: Computer
Second string is lesser
```


C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 1\...

```

<=================================>
Choose one of the following:
1. strlen - Find length of a string
2. strcat - Concatenates 2 strings
3. strcpy - Copies string
4. strcmp - Compares 2 strings
5. strrev - Reverses a string
0. Exit
<=================================>
Enter choice: 5
Enter your string: SachinKumarBaghel
Reversed string: `lehgaBramuKnihcaS`
```

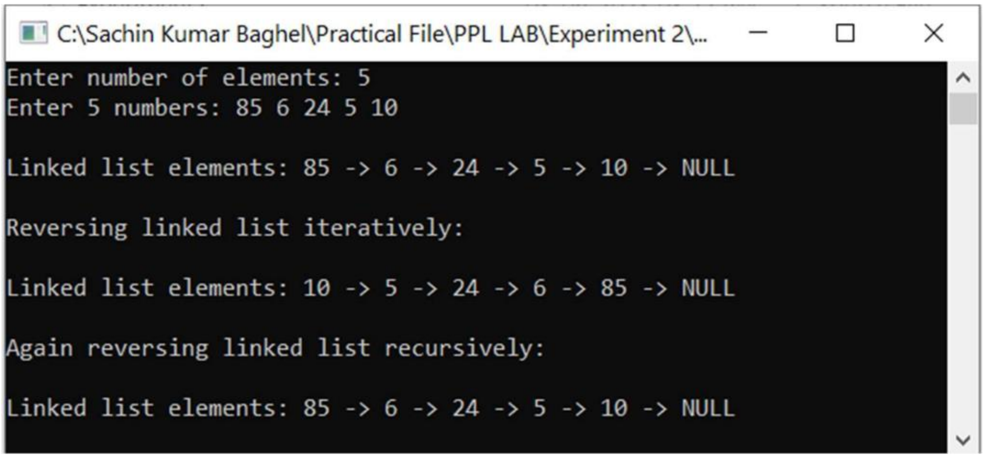
Experiment – 2

Aim - Write a program in C to reverse a linked list iterative and recursive.

Program

```
#include <stdio.h>
#include <stdlib.h>
struct LinkedListNode
{
    int value;
    struct LinkedListNode *next;
};
struct LinkedListNode *newLinkedListNode(int value)
{
    struct LinkedListNode *node = (struct LinkedListNode *)malloc(sizeof(struct
LinkedListNode));
    node->value = value;
    node->next = NULL;
    return node;
}
struct LinkedListNode *createLinkedList(int n)
{
    if (n <= 0)
    {
        return NULL;
    }
    struct LinkedListNode *head, *temp;
    int value;
    scanf("%d", &value);
    head = newLinkedListNode(value);
    temp = head;
    while (--n)
    {
        scanf("%d", &value);
        temp->next = newLinkedListNode(value);
        temp = temp->next;
    }
    return head;
}
void printLinkedList(struct LinkedListNode *head)
{
    struct LinkedListNode *temp = head;
    printf("\nLinked list elements: ");
    while (temp != NULL)
    {
        printf("%d -> ", temp->value);
        temp = temp->next;
    }
    printf("NULL\n");
}
struct LinkedListNode *reverseLinkedListIterative(struct LinkedListNode *head)
{
    struct LinkedListNode *prev = NULL, *curr = head, *next;
    while (curr != NULL)
    {
        next = curr->next;
        curr->next = prev;
        prev = curr;
        curr = next;
    }
    return prev;
}
struct LinkedListNode *reverseLinkedListRecursive(struct LinkedListNode *head)
{
    if (head == NULL || head->next == NULL)
    {
        return head;
    }
    struct LinkedListNode *rest = reverseLinkedListRecursive(head->next);
    head->next->next = head;
    head->next = NULL;
    return rest;
}
int main()
{
    struct LinkedListNode *head;
    int n;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d numbers: ", n);
    head = createLinkedList(n);
    printLinkedList(head);
    printf("\nReversing linked list iteratively:\n");
    head = reverseLinkedListIterative(head);
    printLinkedList(head);
    printf("\nAgain reversing linked list recursively:\n");
    head = reverseLinkedListRecursive(head);
    printLinkedList(head);
    return 0;
}
```

OUTPUT



C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 2\...

```
Enter number of elements: 5
Enter 5 numbers: 85 6 24 5 10

Linked list elements: 85 -> 6 -> 24 -> 5 -> 10 -> NULL

Reversing linked list iteratively:

Linked list elements: 10 -> 5 -> 24 -> 6 -> 85 -> NULL

Again reversing linked list recursively:

Linked list elements: 85 -> 6 -> 24 -> 5 -> 10 -> NULL
```

Experiment – 3

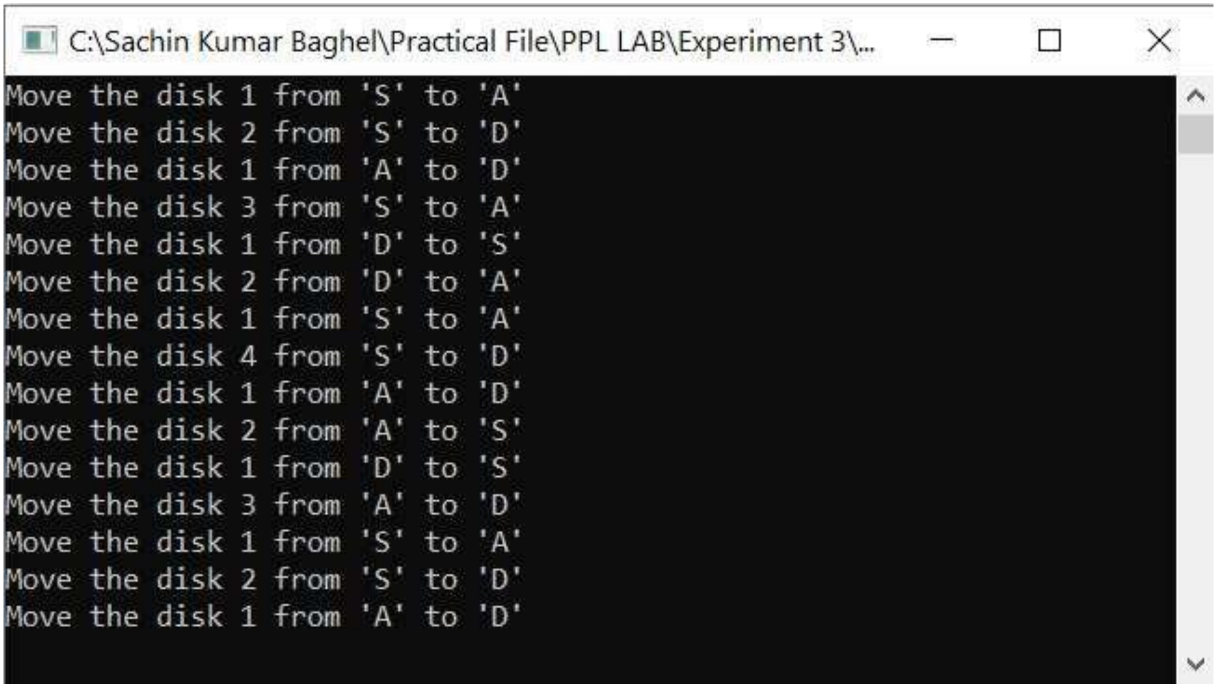
Aim - WAP in C to implement iterative tower of Hanoi.

Program

```
// C Program for Iterative Tower of Hanoi
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <limits.h>

// A structure to represent a stack
struct Stack
{
unsigned capacity;
int top;
int *array;
};

// function to create a stack of given capacity.
struct Stack* createStack(unsigned capacity)
{
    struct Stack* stack =
        (struct Stack*) malloc(sizeof(struct Stack));
    stack -> capacity = capacity;
    stack -> top = -1;
    stack -> array =
        (int*) malloc(stack -> capacity * sizeof(int));
    return stack;
}

// Stack is full when top is equal to the last index
int isFull(struct Stack* stack)
{
return (stack->top == stack->capacity - 1);
}

// Stack is empty when top is equal to -1
int isEmpty(struct Stack* stack)
{
return (stack->top == -1);
}

// Function to add an item to stack. It increases
// top by 1
void push(struct Stack *stack, int item)
{
    if (isFull(stack))
```

```
        return;
    stack -> array[++stack -> top] = item;
}

// Function to remove an item from stack. It
// decreases top by 1
int pop(struct Stack* stack)
{
    if (isEmpty(stack))
        return INT_MIN;
    return stack -> array[stack -> top--];
}

//Function to show the movement of disks
void moveDisk(char fromPeg, char toPeg, int disk)
{
    printf("Move the disk %d from \'%c\' to \'%c\'\n",
        disk, fromPeg, toPeg);
}

// Function to implement legal movement between
// two poles
void moveDisksBetweenTwoPoles(struct Stack *src,
            struct Stack *dest, char s, char d)
{
    int pole1TopDisk = pop(src);
    int pole2TopDisk = pop(dest);

    // When pole 1 is empty
    if (pole1TopDisk == INT_MIN)
    {
        push(src, pole2TopDisk);
        moveDisk(d, s, pole2TopDisk);
    }

    // When pole2 pole is empty
    else if (pole2TopDisk == INT_MIN)
    {
        push(dest, pole1TopDisk);
        moveDisk(s, d, pole1TopDisk);
    }

    // When top disk of pole1 > top disk of pole2
    else if (pole1TopDisk > pole2TopDisk)
    {
        push(src, pole1TopDisk);
        push(src, pole2TopDisk);
        moveDisk(d, s, pole2TopDisk);
    }

    // When top disk of pole1 < top disk of pole2
    else
    {
        push(dest, pole2TopDisk);
        push(dest, pole1TopDisk);
        moveDisk(s, d, pole1TopDisk);
    }
}

//Function to implement TOH puzzle
void tohIterative(int num_of_disks, struct Stack
            *src, struct Stack *aux,
            struct Stack *dest)
{
    int i, total_num_of_moves;
    char s = 'S', d = 'D', a = 'A';

    //If number of disks is even, then interchange
    //destination pole and auxiliary pole
    if (num_of_disks % 2 == 0)
    {
        char temp = d;
        d = a;
        a = temp;
    }
    total_num_of_moves = pow(2, num_of_disks) - 1;

    //Larger disks will be pushed first
    for (i = num_of_disks; i >= 1; i--)
        push(src, i);

    for (i = 1; i <= total_num_of_moves; i++)
    {
        if (i % 3 == 1)
        moveDisksBetweenTwoPoles(src, dest, s, d);

        else if (i % 3 == 2)
        moveDisksBetweenTwoPoles(src, aux, s, a);

        else if (i % 3 == 0)
        moveDisksBetweenTwoPoles(aux, dest, a, d);
    }
}

// Driver Program
int main()
{
    // Input: number of disks
    unsigned num_of_disks = 4;

    struct Stack *src, *dest, *aux;
    // Create three stacks of size 'num_of_disks'
    // to hold the disks
    src  = createStack(num_of_disks);
    aux  = createStack(num_of_disks);
    dest = createStack(num_of_disks);

    tohIterative(num_of_disks, src, aux, dest);
    getch();
    return 0;
}
```

OUTPUT



C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 3\...

```
Move the disk 1 from 'S' to 'A'
Move the disk 2 from 'S' to 'D'
Move the disk 1 from 'A' to 'D'
Move the disk 3 from 'S' to 'A'
Move the disk 1 from 'D' to 'S'
Move the disk 2 from 'D' to 'A'
Move the disk 1 from 'S' to 'A'
Move the disk 4 from 'S' to 'D'
Move the disk 1 from 'A' to 'D'
Move the disk 2 from 'A' to 'S'
Move the disk 1 from 'D' to 'S'
Move the disk 3 from 'A' to 'D'
Move the disk 1 from 'S' to 'A'
Move the disk 2 from 'S' to 'D'
Move the disk 1 from 'A' to 'D'
```

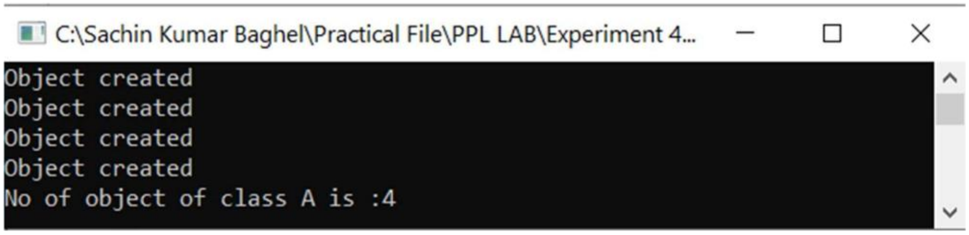
Experiment – 4

Aim - WAP in C++ to count the number of objects of a class with the help of static data member, function and constructor.

Program

```
#include <iostream>
using namespace std;
class A
{
    public:
        static int count;
        A(){
            count++;
            cout << "Object created" << endl;
        }
        static void printMembers(){
            cout << "No of object of class A is :" << count << endl;
        }
};
int A::count = 0;
int main(void){
    A obj1;
    A obj2;
    A obj3;
    A obj4;
    A::printMembers();
    cin.get();
    return 0;
}
```

Output


C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 4...

```
Object created
Object created
Object created
Object created
No of object of class A is :4
```

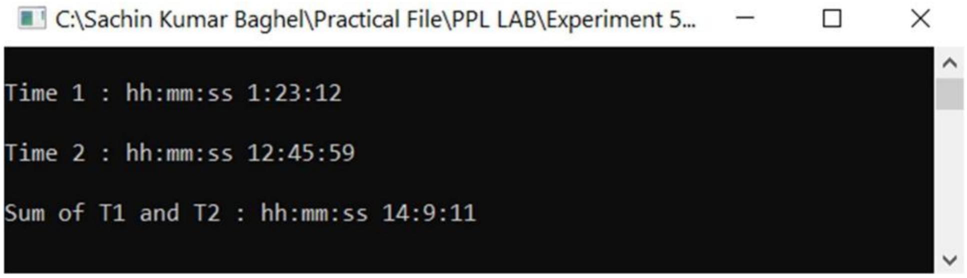
Experiment – 5

Aim - WAP in C++ and Java to declare a class Time with data members mm for minutes, ss for seconds and hh for hours. Define a parameterize constructor to assign time to its objects. Add two-time objects using member function and assign to third objects. Implement all possible cases of time.

C++ Program

```
#include <iostream>
using namespace std;
class Time
{
    public:
        int ss;
        int hh;
        int mm;
        Time(){
            hh = 0;
            mm = 0;
            ss = 0;
        }
        Time(int h, int m,int s){
            hh = h;
            mm = m;
            ss = s;
        }
        Time addTime(Time t){
            Time result(0,0,0);
            long seconds1 = hh * 60 * 60 + mm * 60 + ss;
            long seconds2 = t.hh * 60 * 60 + t.mm * 60 + t.ss;
            long sum = seconds1 + seconds2;
            result.hh = (sum / 3600);
            sum = (sum % 3600);
            result.mm  =  sum  /  60;
            sum = sum % 60;
            result.ss = sum;
            return result;
        }
        void printTime(){
            cout << "hh:mm:ss " << hh << ":" << mm << ":" << ss << endl;
        }
};

int main(void){
    Time t1(1, 23, 12);
    cout << "\nTime 1 : ";
    t1.printTime();
    Time t2(12, 45, 59);
    cout << "\nTime 2 : ";
    t2.printTime();
    Time sum = t1.addTime(t2);
    cout << "\nSum of T1 and T2 : ";
    sum.printTime();
    cin.get();
    return 0;
}
```

OUTPUT

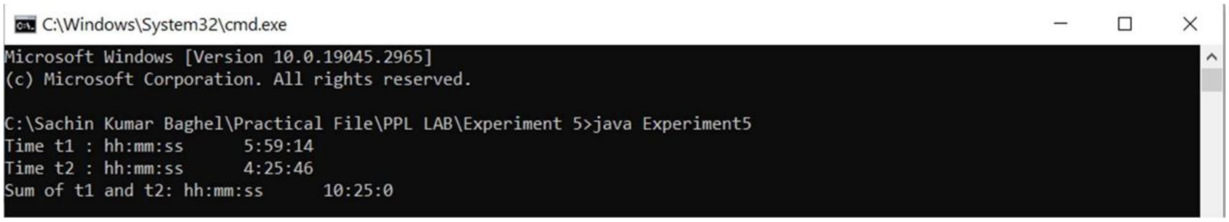

C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 5...

```
Time 1 : hh:mm:ss 1:23:12

Time 2 : hh:mm:ss 12:45:59

Sum of T1 and T2 : hh:mm:ss 14:9:11
```

Java Program

```
public  class  Experiment5  {
    public static class Time {
        public int ss;
        public int hh;
        public int mm;

        public Time(){
            this.hh = 0;
            this.mm = 0;
            this.ss = 0;
        }
        public Time(int hh, int mm, int ss) {
            this.hh = hh;
            this.mm = mm;
            this.ss = ss;
        }
        public void printTime() {
            System.out.println("hh:mm:ss\t" + this.hh + ":" + this.mm + ":" +
this.ss + "\t");
        }
        public Time addTime(Time t) {
            Time result = new Time();
            int seconds1 = hh * 60 * 60 + mm * 60 + ss;
            int seconds2 = t.hh * 60 * 60 + t.mm * 60 + t.ss;
            int sum = seconds1 + seconds2;
            result.hh = (sum / 3600);
            sum = (sum % 3600);
            result.mm = sum / 60;
            sum = sum % 60;
            result.ss = sum;
            return result;
        }
    }
    public static void main(String[] args) {
        Time t1 = new Time(5, 59, 14);
        System.out.print("Time t1 : ");
        t1.printTime();
        Time t2 = new Time(4, 25, 46);
        System.out.print("Time t2 : ");
        t2.printTime();
        Time sum = t1.addTime(t2);
        System.out.print("Sum of t1 and t2: ");
        sum.printTime();
    }
}
```

OUTPUT



```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 5>java Experiment5
Time t1 : hh:mm:ss      5:59:14
Time t2 : hh:mm:ss      4:25:46
Sum of t1 and t2: hh:mm:ss      10:25:0
```

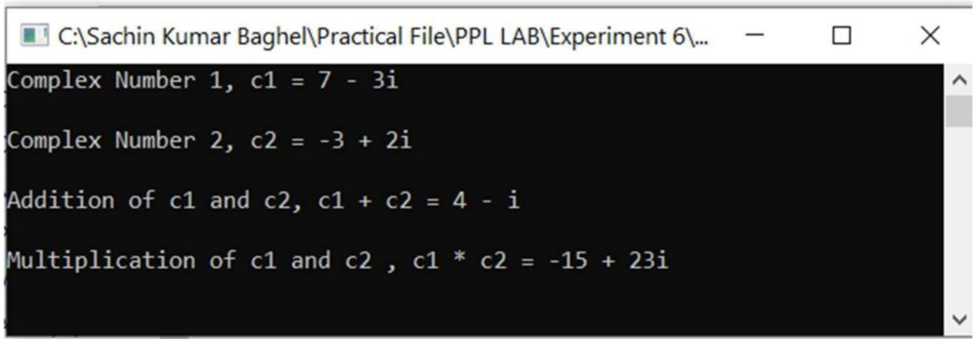
Experiment – 6

Aim - WAP in C++ to define a class Complex to represents set of all complex numbers. Overload ‘+’ operator to add two complex numbers using member function of the class and overload ‘*’ operator to multiply two complex numbers using friend function of the class complex.

Program

```
#include <iostream>
#include <string>
using namespace std;
class Complex
{
public:
    int real, imag;
    Complex(int real = 0, int imag = 0)
    {
        this->real = real;
        this->imag = imag;
    }
    Complex operator+(const Complex &c)
    {
        Complex res;
        res.real = real + c.real;
        res.imag = imag + c.imag;
        return res;
    }
    string getNumberAsString()
    {
        string str = to_string(real);
        str += imag < 0 ? " - " : " + ";
        str += (abs(imag) == 1 ? "" : to_string(abs(imag))) + "i";
        return str;
    }
    friend Complex operator*(const Complex &, const Complex &);
};
Complex operator*(const Complex &c1, const Complex &c2)
{
    Complex res;
    res.real = c1.real * c2.real - c1.imag * c2.imag;
    res.imag = c1.real * c2.imag + c1.imag * c2.real;
    return res;
}
int main()
{
    Complex c1(7, -3), c2(-3, 2);
    cout << "Complex Number 1, c1 = " << c1.getNumberAsString() << endl;
    cout << "\nComplex Number 2, c2 = " << c2.getNumberAsString() << endl;
    Complex c3 = c1 + c2;
    cout << "\nAddition of c1 and c2, c1 + c2 = " << c3.getNumberAsString() << endl;
    Complex c4 = c1 * c2;
    cout << "\nMultiplication of c1 and c2 , c1 * c2 = " << c4.getNumberAsString()
<< endl;
    cin.get();
    return 0;
}
```

OUTPUT



C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 6\...

```
Complex Number 1, c1 = 7 - 3i

Complex Number 2, c2 = -3 + 2i

Addition of c1 and c2, c1 + c2 = 4 - i

Multiplication of c1 and c2 , c1 * c2 = -15 + 23i
```

Experiment – 7

Aim - Implement simple multi-threaded server to perform all mathematics operations parallel in Java.

Program

Server.java

```
import java.io.*;
import java.net.*;
class Server {
    public static void main(String[] args)
    {
        ServerSocket server = null;
        try {
            server = new ServerSocket(1234);
            server.setReuseAddress(true);
            while (true) {
                Socket client = server.accept();
                System.out.println("New client connected"
                                + client.getInetAddress()
                                        .getHostAddress());
                ClientHandler clientSock
                    = new ClientHandler(client);
                new Thread(clientSock).start();
            }
        }
        catch (IOException e) {
            e.printStackTrace();
        }
        finally {
            if (server != null) {
                try {
                    server.close();
                }
                catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
    private static class ClientHandler implements Runnable {
        private final Socket clientSocket;
        public ClientHandler(Socket socket)
        {
            this.clientSocket = socket;
        }
```

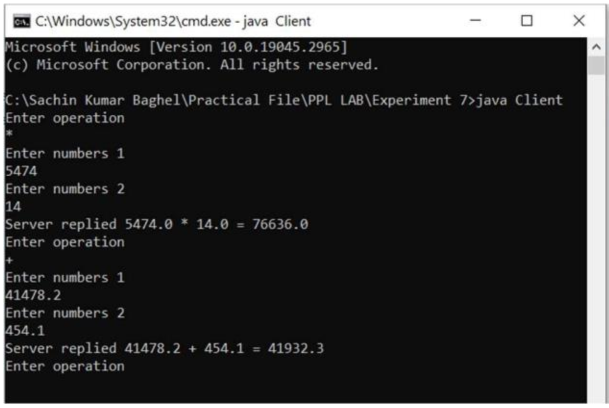
```
public void run()
{
    PrintWriter out = null;
    BufferedReader in = null;
    try {
        out = new PrintWriter(
            clientSocket.getOutputStream(), true);
        in = new BufferedReader(
            new InputStreamReader(
                clientSocket.getInputStream()));

        String line;
        while ((line = in.readLine()) != null) {
            String del = "#";
            String[] temp = line.split(del);
            float x = Float.parseFloat(temp[1]);
            float y = Float.parseFloat(temp[2]);
            char operation = temp[0].charAt(0);
            float result = 0;
            if(operation == '+'){
                result = x + y;
            }else if(operation == '-'){
                result = x- y;
            }else if(operation == '*'){
                result = x * y;
            }else if(operation == '/'){
                result = x / y;
            }else{

            }
            String req = "" + x + " " + operation +" " + y;
            System.out.printf(
                " Sent from the client: %s\n",req
                );
            String res = "" + x +" "+ operation +" "+ y + " = " + result;
            out.println(res);
        }
    }
    catch (IOException e) {
        e.printStackTrace();
    }
    finally {
        try {
            if (out != null) {
                out.close();
            }
            if (in != null) {
                in.close();
                clientSocket.close();
            }
        }
```

```
                catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

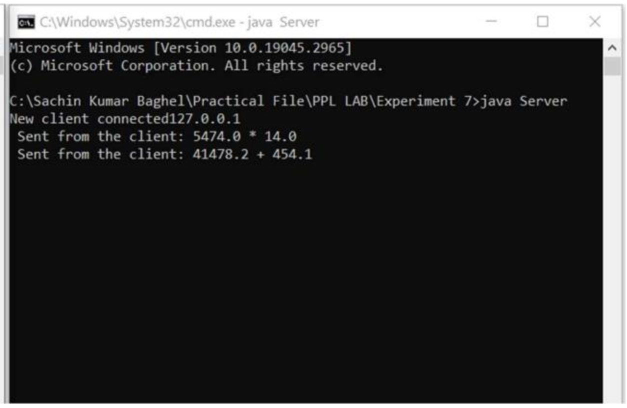
Client.java

```
import java.io.*;
import java.net.*;
import java.util.*;
class Client {
    public static void main(String[] args)
    {
        try (Socket socket = new Socket("localhost", 1234)) {
            PrintWriter out = new PrintWriter(
            socket.getOutputStream(), true);
            BufferedReader in
                = new BufferedReader(new InputStreamReader(
                    socket.getInputStream()));
            Scanner sc = new Scanner(System.in);
            String line = null;

            while (!"exit".equalsIgnoreCase(line)) {

                System.out.println("Enter operation");
                char operation = sc.next().charAt(0);
                System.out.println("Enter numbers 1");
                float x = sc.nextFloat();
                System.out.println("Enter numbers 2");
                float y = sc.nextFloat();
                String req = operation + "#" + x + "#" + y;
                out.println(req);
                out.flush();
                System.out.println("Server replied "
                                + in.readLine());
            }
            sc.close();
        }
        catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

OUTPUT



```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 7>java Client
Enter operation
*
Enter numbers 1
5474
Enter numbers 2
14
Server replied 5474.0 * 14.0 = 76636.0
Enter operation
+
Enter numbers 1
41478.2
Enter numbers 2
454.1
Server replied 41478.2 + 454.1 = 41932.3
Enter operation
```



```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 7>java Server
New client connected127.0.0.1
 Sent from the client: 5474.0 * 14.0
 Sent from the client: 41478.2 + 454.1
```

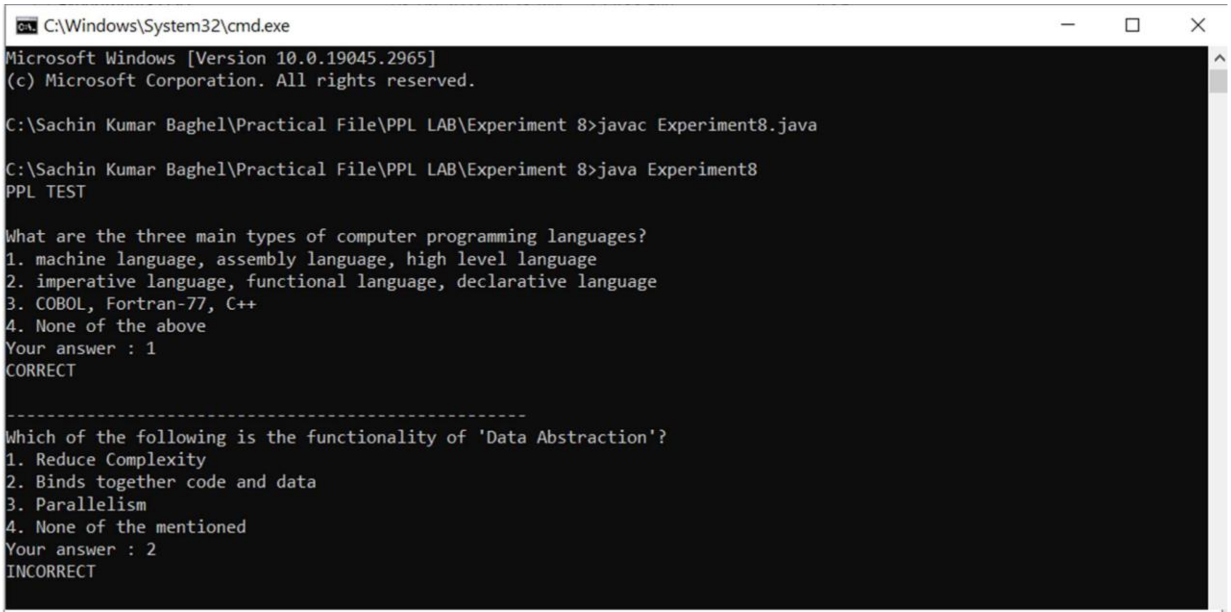
Experiment – 8

Aim - Write a program to prepare a list of 10 questions and their answers

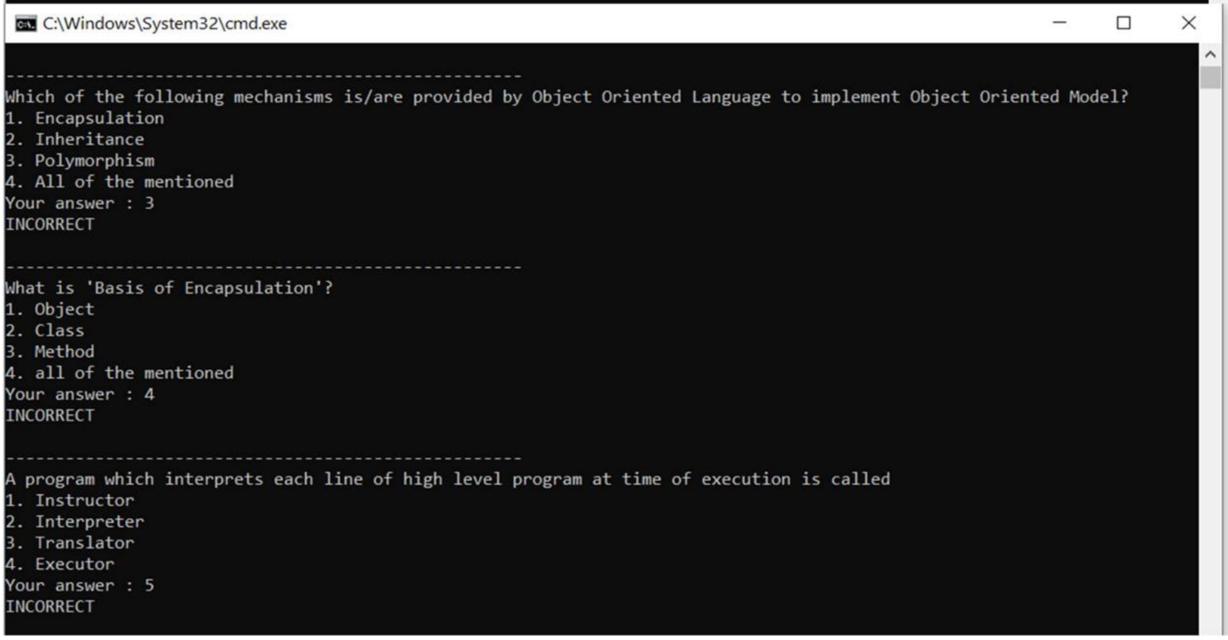
Program

```
import java.io.*;
import java.util.*;
public class Experiment8{
    public static class Question{
        String ques;
        String[] options;
        int answer;

        public Question() {
            this.ques = "";
            this.options = new String[4];
            this.answer = 1;
        }

        public Question(String ques, String[] options,int answer){
            this.ques = ques;
            this.options = options;
            this.answer = answer;
        }

        public boolean checkAnswer(int answer) {
            if (answer == this.answer)
                return true;
            else
                return false;
        }

        public void display(){
            System.out.println(this.ques);
            for (int i = 0; i < this.options.length; i++) {
                System.out.println((i + 1) + ". " + this.options[i]);
            }
        }
    }

    public static void main(String[] args){
        Scanner scn = new Scanner(System.in);
        Question[] list = new Question[10];
        list[0] = new Question(
            "What are the three main types of computer programming languages?",
                new String[] {
                "machine language, assembly language, high level language",
                "imperative language, functional language, declarative language",
                "COBOL, Fortran-77, C++",
                "None of the above"},
            1
        );
        list[1] = new Question(
            "Which of the following is the functionality of \'Data Abstraction\'?",
                new String[] {
                "Reduce Complexity",
                "Binds together code and data",
                "Parallelism",
                "None of the mentioned"},
            1
        );
        list[2] = new Question(
            "Which of the following mechanisms is/are provided by Object Oriented
Language to implement Object Oriented Model?",
                new String[] {
                "Encapsulation",
                "Inheritance",
                "Polymorphism",
                "All of the mentioned"},
            4
        );
        list[3] = new Question(
            "What is \'Basis of Encapsulation\'?",
                new String[] {
                "Object",
                "Class",
                "Method",
                "all of the mentioned"},
            1
        );
        list[4] = new Question(
            "A program which interprets each line of high level program at time of
execution is called",
                new String[] {
                "Instructor",
                "Interpreter",
                "Translator",
                "Executor"},
            2
        );
        list[5] = new Question(
            "Programming language \'BASIC\' is used for the",
                new String[] {
                "Beginners",
                "commercial programs",
                "household user interface",
                "student applications"},
            1
        );
        list[6] = new Question(
            "\'object program\' is also called",
                new String[] {
                "program code",
                "machine code",
                "assembler",
                "compiler"},
            1
        );
        list[7] = new Question(
            "In programming language COBOL, symbol of \'//\' is used instead of",
                new String[] {
                    "unmarked variable",
                    "unmarked strings",
                    "remarks",
                    "marked structure"
                },
            3
        );
        list[8] = new Question(
            "In programming language BASIC, area is calculated as",
                new String[] {
                "100 Area=Width*Length",
                "100 Area : = Width*Length",
                "Area : = Width*Length",
                "Area 100 : length*width"},
            1
        );
        list[9] = new Question(
            "Loop which is tested at least once in case condition does not fulfilled
is classified as",
                new String[] {
                "FOR loop",
                "GO loop",
                "REPEAT loop",
                "REPEAT UNTIL loop"},
            4
        );
        System.out.println("PPL TEST\n");
        for(int i = 0; i < list.length;i++){
            list[i].display();
            System.out.print("Your answer : ");
            int answer = scn.nextInt();
            System.out.println(list[i].checkAnswer(answer) ? "CORRECT" :
"INCORRECT");
            System.out.print("\n___________________________________________________
\n");
        }

    }
}
```

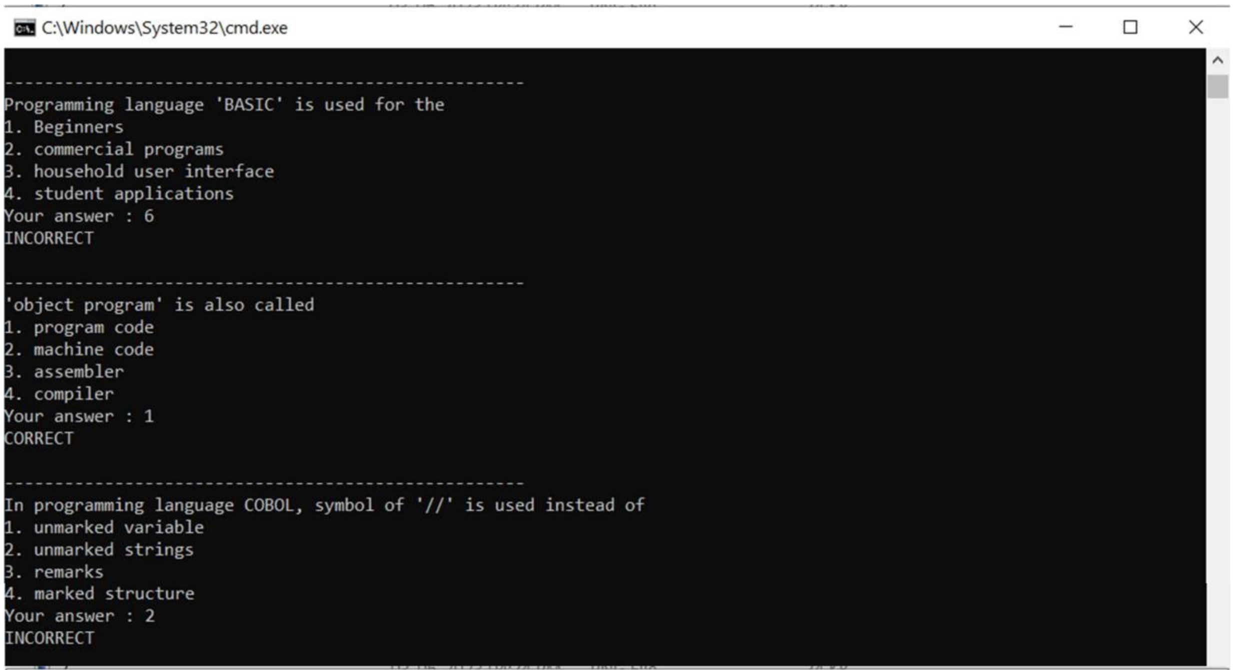
OUTPUT



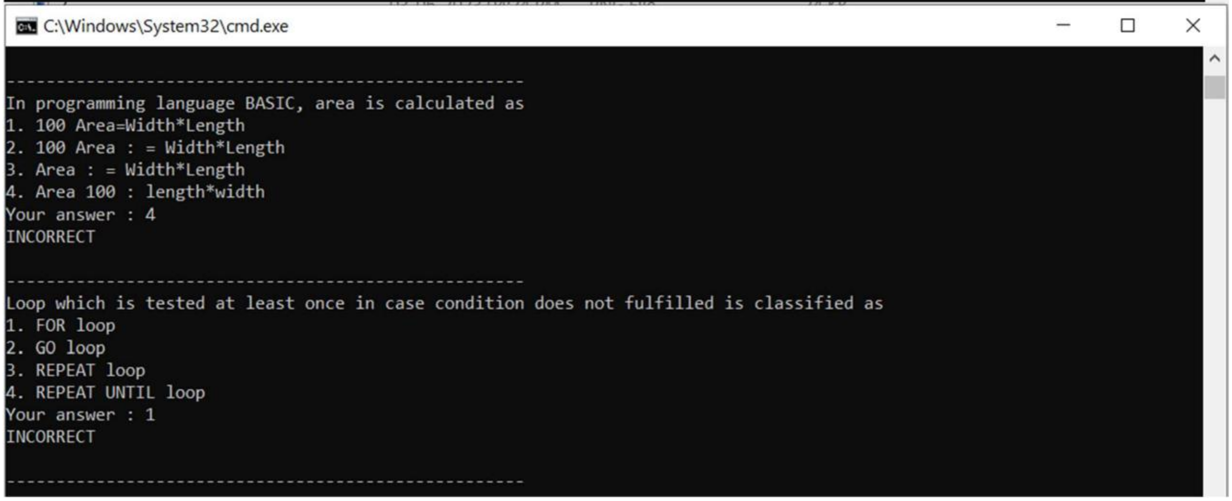
```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 8>javac Experiment8.java

C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 8>java Experiment8
PPL TEST

What are the three main types of computer programming languages?
1. machine language, assembly language, high level language
2. imperative language, functional language, declarative language
3. COBOL, Fortran-77, C++
4. None of the above
Your answer : 1
CORRECT

----------------------------------------------------
Which of the following is the functionality of 'Data Abstraction'?
1. Reduce Complexity
2. Binds together code and data
3. Parallelism
4. None of the mentioned
Your answer : 2
INCORRECT
```



```
----------------------------------------------------
Which of the following mechanisms is/are provided by Object Oriented Language to implement Object Oriented Model?
1. Encapsulation
2. Inheritance
3. Polymorphism
4. All of the mentioned
Your answer : 3
INCORRECT

----------------------------------------------------
What is 'Basis of Encapsulation'?
1. Object
2. Class
3. Method
4. all of the mentioned
Your answer : 4
INCORRECT

----------------------------------------------------
A program which interprets each line of high level program at time of execution is called
1. Instructor
2. Interpreter
3. Translator
4. Executor
Your answer : 5
INCORRECT
```



```
----------------------------------------------------
Programming language 'BASIC' is used for the
1. Beginners
2. commercial programs
3. household user interface
4. student applications
Your answer : 6
INCORRECT

----------------------------------------------------
'object program' is also called
1. program code
2. machine code
3. assembler
4. compiler
Your answer : 1
CORRECT

----------------------------------------------------
In programming language COBOL, symbol of '//' is used instead of
1. unmarked variable
2. unmarked strings
3. remarks
4. marked structure
Your answer : 2
INCORRECT
```



```
----------------------------------------------------
In programming language BASIC, area is calculated as
1. 100 Area=Width*Length
2. 100 Area : = Width*Length
3. Area : = Width*Length
4. Area 100 : length*width
Your answer : 4
INCORRECT

----------------------------------------------------
Loop which is tested at least once in case condition does not fulfilled is classified as
1. FOR loop
2. GO loop
3. REPEAT loop
4. REPEAT UNTIL loop
Your answer : 1
INCORRECT

----------------------------------------------------
```

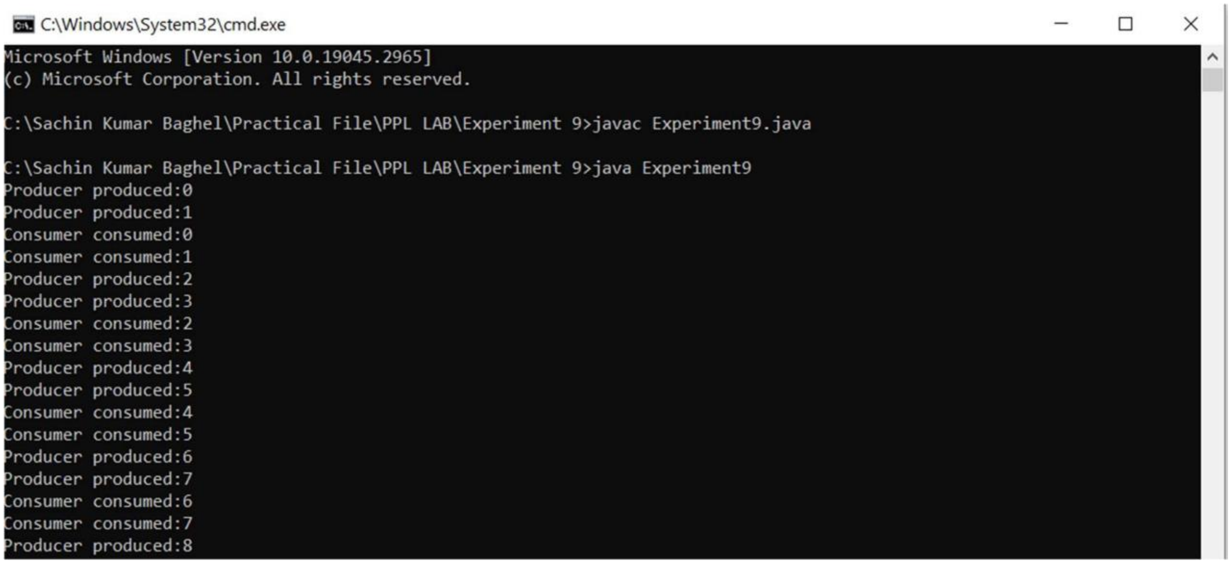
Experiment – 9

Aim - Implement producer-consumer problem using threads.

Program

```
import java.util.LinkedList;
public class Experiment9 {
    public static void main(String[] args)
        throws InterruptedException
    {
        final ProducerConsumer pc = new ProducerConsumer();
        Thread t1 = new Thread(new Runnable() {
            @Override
            public void run()
            {
                try {
                    pc.produce();
                }
                catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        });
        Thread t2 = new Thread(new Runnable() {
            @Override
            public void run()
            {
                try {
                    pc.consume();
                }
                catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        });
        t1.start();
        t2.start();
        t1.join();
        t2.join();
    }
    public static class ProducerConsumer {

        LinkedList<Integer> list = new LinkedList<>();
        int capacity = 2;
        public void produce() throws InterruptedException
        {
            int value = 0;
            while (true) {
                synchronized (this)
                {
                    while (list.size() == capacity)
                    wait();
                    System.out.println("Producer produced:"+ value);
                    list.add(value++);
                    notify();
                    Thread.sleep(1000);
                }
            }
        }
        public void consume() throws InterruptedException
        {
            while (true) {
                synchronized (this)
                {
                    while (list.size() == 0)
                    wait();
                    int val = list.removeFirst();
                    System.out.println("Consumer consumed:"+ val);
                    notify();
                    Thread.sleep(1000);
                }
            }
        }
    }
}
```

OUTPUT



```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 9>javac Experiment9.java

C:\Sachin Kumar Baghel\Practical File\PPL LAB\Experiment 9>java Experiment9
Producer produced:0
Producer produced:1
Consumer consumed:0
Consumer consumed:1
Producer produced:2
Producer produced:3
Consumer consumed:2
Consumer consumed:3
Producer produced:4
Producer produced:5
Consumer consumed:4
Consumer consumed:5
Producer produced:6
Producer produced:7
Consumer consumed:6
Consumer consumed:7
Producer produced:8
```