

# **KG COLLEGE OF ARTS AND SCIENCE**

(Affiliated to Bharathiar University & Accredited by NAAC)

KGiSL Campus, Saravanampatti, Coimbatore- 641035.

## ***KGCAJ***



## **DEPARTMENT OF COMPUTER APPLICATIONS**

### **CORE LAB: PROGRAMMING LAB – C++**

## **LAB MANUAL**



## **AUTHORS**

Dr.D.Kavitha  
Associant Professor  
KG College of Arts and Science

Mrs.G.Priyadarshini  
Assistant Professor  
KG College of Arts and Science

Mrs. D. Jeyarani  
Assistant Professor  
KG College of Arts and Science



## **C++ PROGRAMMING LAB**

### **SYLLABUS** **CORE LAB 1 – C++ PROGRAMMING LAB**

<b>Subject Code</b>	<b>23P</b>
<b>Internal Marks</b>	<b>40</b>
<b>External Marks</b>	<b>60</b>
<b>Total Marks</b>	<b>100</b>

#### **COURSE OBJECTIVE**

The main objectives of this course are to:

1. Impart knowledge of object oriented programming concepts and implement them in C++
2. Enable to differentiate procedure oriented and object-oriented concepts.
3. Equip with the knowledge of concept of Inheritance so that learner understands the need of inheritance.
4. Explain the importance of data hiding in object oriented programming

#### **COURSE OUTCOME**

1. Define the different programming paradigm such as procedure oriented and object oriented programming methodology and conceptualize elements of OO methodology.
2. Illustrate and model real world objects and map it into programming objects for a legacy system.
3. Identify the concepts of inheritance and its types and develop applications using overloading features.
4. Discover the usage of pointers with classes.
5. Explain the usage of Files, templates and understand the importance of exception Handling.



## **C++ PROGRAMMING LAB**

### **LAB SYLLABUS**

1. Write a C++ Program to create a class to implement the data structure STACK. Write a constructor to initialize the TOP of the STACK. Write a member function PUSH() to insert an element and member function POP() to delete an element check for overflow and underflow conditions.
2. Write a C++ Program to create a class ARITHMETIC which consists of a FLOAT and an INTEGER variable. Write member functions ADD (),SUB(), MUL(), DIV() to perform addition, subtraction, multiplication, division respectively. Write a member function to get and display values.
3. Write a C++ Program to read an integer number and find the sum of all the digits until it reduces to a single digit using constructors, destructors and inline member functions.
4. Write a C++ Program to create a class FLOAT that contains one float data member. Overload all the four Arithmetic operators so that they operate on the object FLOAT.
5. Write a C++ Program to create a class STRING. Write a Member Function to initialize, get and display strings. Overload the operators ++ and == to concatenate two Strings and to compare two strings respectively.
6. Write a C++ Program to create class, which consists of EMPLOYEE Detail like E\_Number, E\_Name, Department, Basic, Salary, Grade. Write a member function to get and display them. Derive a class PAY from the above class and write a member function to calculate DA, HRA and PF depending on the grade.
7. Write a C++ Program to create a class SHAPE which consists of two VIRTUALFUNCTIONS Calculate\_Area() and Calculate\_Perimeter() to calculate area and perimeter of various figures. Derive three classes SQUARE, RECTANGLE, TRIANGLE from class Shape and Calculate Area and Perimeter of each class separately and display the result.
8. Write a C++ Program to create two classes each class consists of two private variables, a integer and a float variable. Write member functions to get and display them. Write a FRIEND Function common to both classes, which takes the object of above two classes as arguments and the integer and float values of both objects separately and display the result.
9. Write a C++ Program using Function Overloading to read two Matrices of different Data Types such as integers and floating point numbers. Find out the sum of the above two matrices separately and display the sum of these arrays individually.
10. Write a C++ Program to check whether the given string is a palindrome or not using Pointers.
11. Write a C++ Program to create a File and to display the contents of that file with linenumbers.
12. Write a C++ Program to merge two files into a single file.



## **C++ PROGRAMMING LAB**

### **HARDWARE & SOFTWARE REQUIREMENTS**

#### **HARDWARE REQUIREMENTS:**

- Pentium Dual Core @ 2.70 Ghz and above with 2GB RAM.
- 160 GB HARD Disk.
- Monitor 1024\* 768 color.
- Optical Mouse & Keyboard.

#### **SOFTWARE REQUIREMENTS:**

- Windows Operating System 7 / 8 / 8.1 / 10.
- Turbo C++ 3.2.
- No Additional Requirements.



## C++ PROGRAMMING LAB

### LIST OF EXPERIMENTS

S.NO.	TITLE	PAGE.NO.
1.	Write a C++ Program to create a class to implement the data structure STACK. Write a constructor to initialize the TOP of the STACK. Write a member function PUSH() to insert an element and member function POP() to delete an element check for overflow and underflow conditions.	1
2.	Write a C++ Program to create a class ARITHMETIC which consists of a FLOAT and an INTEGER variable. Write member functions ADD (),SUB(), MUL(), DIV() to perform addition, subtraction, multiplication, division respectively. Write a member function to get and display values.	9
3.	Write a C++ Program to read an integer number and find the sum of all the digits until it reduces to a single digit using constructors, destructors and inline member functions.	13
4.	Write a C++ Program to create a class FLOAT that contains one float data member. Overload all the four Arithmetic operators so that they operate on the object FLOAT.	18
5.	Write a C++ Program to create a class STRING. Write a Member Function to initialize, get and display strings. Overload the operators ++ and == to concatenate two Strings and to compare two strings respectively.	21
6.	Write a C++ Program to create class, which consists of EMPLOYEE Detail like E_Number, E_Name, Department, Basic, Salary, Grade. Write a member function to get and display them. Derive a class PAY from the above class and write a member function to calculate DA, HRA and PF depending on the grade.	24
7.	Write a C++ Program to create a class SHAPE which consists of two VIRTUALFUNCTIONS Calculate_Area() and Calculate_Perimeter() to calculate area and perimeter of various figures. Derive three classes SQUARE, RECTANGLE, TRIANGLE from class Shape and Calculate Area and Perimeter of each class separately and display the result.	27



## C++ PROGRAMMING LAB

8.	Write a C++ Program to create two classes each class consists of two private variables, a integer and a float variable. Write member functions to get and display them. Write a FRIEND Function common to both classes, which takes the object of above two classes as arguments and the integer and float values of both objects separately and display the result.	32
9.	Write a C++ Program using Function Overloading to read two Matrices of different Data Types such as integers and floating point numbers. Find out the sum of the above two matrices separately and display the sum of these arrays individually.	35
10.	Write a C++ Program to check whether the given string is a palindrome or not using Pointers.	39
11.	Write a C++ Program to create a File and to display the contents of that file with line numbers.	42
12.	Write a C++ Program to merge two files into a single file.	45

**EXPERIMENT 1****AIM:1**

Write a C++ Program to create a class to implement the data structure STACKS. Write a constructor to initialize the TOP of the STACK. Write a member function PUSH () to insert an element and member function POP () to delete an element check for overflow and underflow condition.

**PROCEDURE:**

- Step 1: Start the process.
- Step 2: Include the header files iostream.h, conio.h and stdlib.h.
- Step 3: Create a class stack with member a[10] and TOP as integer.
- Step 4: Using constructor initialize the variable TOP as 0.
- Step 5: The member function PUSH() is used to get input from the user for inserting an element into the stack.
- Step 6: The member function POP() is used to delete an element from the stack.
- Step 7: The member function display () is used to display the number of elements available in the stack.
- Step 8: In main function, create an object S for the class stack.
- Step 9: Display stack operation with four choices.
- Step 10: Getting the option from the user and perform stack operations using switch statement.
- Step 11: Stop the process.

**SOURCE CODE:**

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
class stack
{
    int a[10],top;
public:
    stack()
    {
        top=0;
    }
    void push(void);
    void pop(void);
    void display(void);
};
void stack :: push()
{
    int i,n;
    cout<<"\n enter no. of elements:"; cin>>n;
    cout<<"\n enter the elements";
    for(i=0;i<n;i++)
    {
        if(top<10)
```



## C++ PROGRAMMING LAB

```
{
cin>>a[top++];
}
else
{
cout<<"\n Stack is full";
break;
}
}
getch();
}
void stack :: pop()
{

if(top==0)
{
cout<<"\n Stack is empty";
}
else
{
top--;
cout<<a[top]<<" deleted";
}
getch();
}
void stack :: display()
{
int i;
cout<<"\n content of stack";
for(i=top-1;i>=0;i--)
cout<<"\n"<<a[i]; getch();
}
void main()
{
stack s;
int opt;
do
{
clrscr();
cout<<"Stack Operations"; cout<<"\n.....";
cout<<"\n 1. Push";
cout<<"\n 2. Pop";
cout<<"\n 3. Display";
cout<<"\n 4. Exit";
cout<<"\n Enter your option : ";
cin>>opt;
switch(opt)
{
case 1:
s.push();
```

## **C++ PROGRAMMING LAB**

```
        break;
case 2:
s.pop();
break;
case 3:
s.display();
break;
case 4:
exit(1);
}
}while(opt<=3); getch();
}
```

### **OUTPUT:**

#### STACK OPERATIONS

- 1.PUSH
- 2.POP
- 3.DISPLAY
- 4.EXIT

ENTER YOUR OPTION: 1

Enter no. of elements: 5

Enter the elements:

20

30

40

50

60

ENTER YOUR OPTION: 2

60 is deleted

ENTER YOUR OPTION: 3

The contents of stack

50

40

30

20

### **CONCLUSION:**

The above program has been verified and executed successfully.

## **DEPARTMENT OF COMPUTER APPLICATIONS**



## **C++ PROGRAMMING LAB**

### **EXPERIMENT 2**

#### **AIM:**

To write a C++ program to create a class ARITHMETIC which consists of a FLOAT and an INTEGER variable. Write member functions ADD(), SUB(), MUL(), DIV() to perform addition, subtraction, multiplication, division respectively. Write a member function to get and display values.

#### **PROCEDURE:**

- Step 1: Start the process.
- Step 2: Include the header files iostream.h and conio.h.
- Step 3: Create a class arithmetic with members a as integer and b as float data type and create a member function getdata() for getting the input from the user.
- Step 4: Likewise, create five member functions and perform the each operation on the data on ADD() for Addition, SUB() for Subtraction, MUL() for Multiplication, DIV() for Division and MOD() for Modulo Division.
- Step 5: In the main function create the object ao for class arithmetic.
- Step 6: Display the output which is processed by the five Member functions of class Arithmetic using the object ao and display the result.
- Step 7: Stop the process.

#### **SOURCE CODE:**

```
#include<iostream.h>
#include<conio.h>
class arithmetic
{
    int a; float b;
public:
    void getdata()
    {
        cout<<"\n Enter a value:"; cin>>a;
        cout<<"\n Enter b value:"; cin>>b;
    }
    void add() // Addition
    {
        cout<<"\n\n Addition: a+b="<<a+b;
    }
    void sub() // Subtraction
    {
        cout<<"\n\n Subtraction: a-b="<<a-b;
    }
    void mul() // Multiplication
```



## **C++ PROGRAMMING LAB**

```
{
cout<<"\n\n Multiplication: a*b="<<a*b;
}
void div() // Division
{
cout<<"\n\n Division: a/b="<<a/b;
}
void mod() // Modulo division
{
cout<<"\n\n Modulo Division="<<a%(int)b;
}
};

void main()
{
clrscr();
cout<<"\n\n\t\t\t ARITHMETIC OPERATIONS\n"; arithmetic ao;
ao.getdata();
ao.add();
ao.sub();
ao.mul();
ao.div();
ao.mod();
getch();
}
```

### **OUTPUT:**

#### **ARITHMETIC OPERATIONS**

Enter a value: 4  
Enter b value: 2.2  
Addition a+b=6.2  
Subtraction a-b=1.8  
Multiplication a\*b=8.8  
Division a/b=1.81812  
Modulo Division=0

### **CONCLUSION:**

The above program has been verified and executed successfully.

## **DEPARTMENT OF COMPUTER APPLICATIONS**



## C++ PROGRAMMING LAB

### EXPERIMENT 3

#### AIM:

To write a C++ Program to read an integer number and to find the sum of all the digits until it reduces to a single digit using constructors, destructors and inline member functions.

#### PROCEDURE:

Algorithm:

- Step 1 : Start the process
- Step 2 : Include the header files iostream.h and conio.h
- Step 3 : Create a class number and under the scope create two members no and sum as integer data type.
- Step 4 : In class create constructor and get data from the user for the variable no.
- Step 5 : Also create a destructor for re-allocating memory space of the object at last.
- Step 6 : Using inline function sum the all individual digits and store the result in the variable result.
- Step 7 : In add member function, use the while loop to check the no is greater than 0 and sum the all individual digits.
- Step 8 : If sum is greater than 9 interchange no into sum and sum into no. Step 9 : Continue the while loop until no become 0 and display the sum. Step 9 : In the main function create a object n and display the result.
- Step 10 : Stop the process.

#### SOURCE CODE:

```
: #include<iostream.h>
#include<conio.h>
class number
{

int no, sum;
public:
void add(void);
inline void display(int); number()
{
sum=0;
}

cout<<"\n\n Enter a number:"; cin>>no;
}
~number()
{
}

};
```



## **C++ PROGRAMMING LAB**

```
inline void number::display(int result)
{
    cout<<"\n\n Sum of Individual Digits="<<result;
}

void number::add()
{
    loop: while(no>0)
    {
        sum=sum+no%10; no=no/10;
    }
    if(sum>9)
    {
        no=sum; sum=0; goto loop;
    }
    display(sum);
}

void main()
{
    clrscr();
    number n;
    n.add();
    getch();
}
```

### **OUTPUT:**

Enter a number: 345  
Sum of Individual Digits=3

### **CONCLUSION:**

The above program has been verified and executed successfully.



## C++ PROGRAMMING LAB

### EXPERIMENT 4

#### AIM:

Write a C++ program to create a class FLOAT which contains one float data member and overload all arithmetic operator to operate on the object FLOAT.

#### PROCEDURE:

- Step 1: Start the process.
- Step 2: Include the header file iostream.h and conio.h. Step 3: Create a class FLOAT with one data member.
- Step 4: Using the constructor get the input from the user for float number.
- Step 5: Member function operator() used to do arithmetic operation and overload all the arithmetic operators addition, subtraction, multiplication and division.
- Step 6: In main function, create two objects and use the arithmetic operators to perform calculations and display the result.
- Step 7: Stop the process.

Source Code:

```
#include<iostream.h>
#include<conio.h>
class FLOAT
{
float a; public: FLOAT()
{

cout<<"\n\n Enter the float number:"; cin>>a;
}
void operator+(FLOAT x)
{
cout<<"\n\n\t ADDITION OF TWO NUMBERS:"<<a+x.a;

}
void operator-(FLOAT x)
{
cout<<"\n\n\t SUBTRACTION OF TWO NUMBERS:"<<a-x.a;
}
void operator*(FLOAT x)
{
cout<<"\n\n\t MULTIPLICATION OF TWO NUMBERS:"<<a*x.a;
}
void operator/(FLOAT x)
{
cout<<"\n\n\t DIVISION OF TWO NUMBERS:"<<a/x.a;
}
};
```



## **C++ PROGRAMMING LAB**

void main()

```
{  
clrscr();  
FLOAT f1,f2;  
f1+f2; f1-f2; f1*f2; f1/f2; getch();  
}
```

### **OUTPUT:**

Enter the float number: 23.4  
Enter the float number: 42.3  
ADDITION OF TWO NUMBERS: 65.7  
SUBTRACTION OF TWO NUMBERS: -18.9  
MULTIPLICATION OF TWO NUMBERS: 989.819946  
DIVISION OF TWO NUMBERS:0.553191

### **CONCLUSION:**

The above program has been verified and executed successfully.

## **EXPERIMENT 5**

### **AIM:**

Write a C++ program to create a class STRING . Write a member function to initiation get and display string overload the operation ++and == to concatenate two string and to compare two string respectively.

### **PROCEDURE:**

- Step 1: Start the process.
- Step 2: Include the header files iostream.h, conio.h and string.h. Step 3: Create a class string with member a [50] as character.
- Step 4: A member function get() is used to get input from the user for the string variable a.
- Step 5: Member function display() is used to display the length of the string a.
- Step 6: The operator == is used to compare two strings with the help of if statement and strcmp() function.
- Step 7: If the strcmp(a,x.A)=0,the given string are equal otherwise unequal .
- Step 8: The operator + is used to concatenate two strings using the built in function strcat().
- Step 9: In main() function declare two objects namely s1 and s2.

## **DEPARTMENT OF COMPUTER APPLICATIONS**





## C++ PROGRAMMING LAB

Step 10: Call the member function get() and display() to get input and print the length of the input of two objects.

Step 11: Use the operators == and + and display the result. Step 12: Stop the process.

### SOURCE CODE:

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class STRING
{
char a[50]; public:
void get()
{
cout<<"\n\n Enter the String:\t"; cin>>a;
}
void display()
{
cout<<"The length of the string is :\t"<<a<<"\t"<<strlen(a);
}
void operator==(STRING x) // comparing two strings
{
cout<<"\n\n\t\t String Comparision\n"; if((strcmp(a,x.a))==0)
cout<<"\n\t\t Strings are equal\n"; else
cout<<"\n\t\t Strings are unequal\n";
}
void operator+(STRING x) // combining two strings
{
cout<<"\n Concatenated String:\t"; cout<<strcat(a,x.a);
cout<<"\n\n\n Length of the Concatenated string is:\t"<<strlen(a);

}
};

void main()
{
clrscr(); STRING s1,s2;
s1.get(); cout<<"\n"; s2.get();
cout<<"\n"; s1.display();
cout<<"\n\n"; s2.display();
s1==s2;
s1+s2; getch();
}
```

### OUTPUT:



## **C++ PROGRAMMING LAB**

Enter the String: COMPUTER  
Enter the String: SCIENCE  
The length of the string is : COMPUTER 8  
The length of the string is : SCIENCE 7  
String Comparison : Strings are not equal

Concatenated String: COMPUTERSCIENCE Length of the Concatenated string is: 15

Enter the String: COMPUTER  
Enter the String: COMPUTER  
The length of the string is : COMPUTER 8  
The length of the string is : COMPUTER 8  
String Comparison Strings are equal

Concatenated String: COMPUTERCOMPUTER Length of the Concatenated string is: 16

### **CONCLUSION:**

The above program has been verified and executed successfully.

## **EXPERIMENT 6**

### **AIM:**

Write a C++ Program to create class, which consists of EMPLOYEE Detail like E\_Number, E\_Name, Department, Basic, Salary, Grade. Write a member function to get and display them. Derive a class PAY from the above class and write a member function to calculate DA, HRA and PF depending on grade.

### **PROCEDURE:**

- Step 1 : Start the process.
- Step 2 : Include the header files iostream.h and conio.h.
- Step 3 : Create a user-define method line() to draw a solid line.
- Step 4 : Create a class with name emp and declare the member variable eno, ename, dept and basic pay.
- Step 5 : Using the member function getdata() to get the input from the user for eno, ename, dept and basic pay.
- Step 6 : Create a derived class pay from the base class emp and declare the member variables da, np, gs, hra and pf as float.
- Step 7 : With the help of calc() member function get the grade and calculate da, hra, pf, gross salary and net pay.
- Step 8 : Print all the details of employee with line() function for pay slip. Step 9 : In main() function declare an object p of class pay.
- Step 10 : Call the getdata() and calc() to and get and display the pay slip of the employee.
- Step 11 : Stop the process.

### **SOURCE CODE:**

```
#include<iostream.h>
```

**DEPARTMENT OF COMPUTER APPLICATIONS**



## C++ PROGRAMMING LAB

```
#include<conio.h>
void line()
{
cout.width(80); cout.fill('_');
}
class emp
{
public:
int eno;
char gr, ename[25], dept[20]; float bs;
void getdata()
{
cout<<"\n Enter the employee no: "; cin>>eno;
cout<<"\n Enter the employee name: "; cin>>ename;
cout<<"\n Enter the employee department: "; cin>>dept;
cout<<"\n Enter the basic pay: "; cin>>bs;
cout<<"\n Enter the grade for salary(a/b): "; cin>>gr;
}
};
class pay:public emp
{
float da,np,gs,hra,pf; public:
void calc()
{
if(gr=='a' || gr=='A')
{
da=bs*0.08; hra=bs*0.2; pf=bs*0.06;
}
else if(gr=='b' || gr=='B')
{
da=bs*0.06; hra=bs*0.15; pf=bs*0.05;
}
gs=bs+da+hra; np=gs-pf;
cout<<"\n\t\t\t EMPLOYEE PAYS LIP"; cout<<"\n\n\n ENO:"<<eno; cout<<"\n\n\n ENAME:"<<ename;
cout<<"\n\n\n DEPARTMENT:"<<dept; cout<<"\n\n\n JULY 2021"; cout<<"\n\n";
line(); cout<<"\n\n"; cout.precision(2);
cout.setf(ios::fixed,ios::floatfield); cout<<"BASIC PAY:"<<bs;
cout<<"\t DA:"<<da; cout<<"\t HRA:"<<hra; cout<<"\t PF:"<<pf; cout<<"\n\n";
line(); cout<<"\n\n";
cout<<"GROSS SALARY:"<<gs;
cout<<"\n\n";
cout<<"\n\n NET SALARY:"<<np; cout<<"\n\n";
line();
cout<<"\n\n\t\t\t THANK YOU";
}
};
void main()
```

## DEPARTMENT OF COMPUTER APPLICATIONS



## **C++ PROGRAMMING LAB**

```
{  
  
clrscr(); pay p; p.getdata();  
p.calc();  
getch();  
}
```

### **OUTPUT:**

Enter the employee no:123 Enter the employee name:ERIC  
Enter the employee department:CS Enter the basic pay:10000  
Enter the grade for salary(a/b):A

EMPLOYEE PAYSALIP  
ENO :123 ENAME :ERIC  
DEPARTMENT :CS JULY 2021  
BASIC PAY : 10000 DA : 800 HRA : 2000 PF : 600  
GROSS SALARY : 12800 NET SALARY : 12200  
THANK YOU

### **CONCLUSION:**

The above program has been verified and executed successfully.

## **EXPERIMENT 7**

### **AIM:**

Write a C++ Program to create a class SHAPE which consists of two VIRTUAL FUNCTIONS Calculate\_Area() and Calculate\_Perimeter() to calculate area and perimeter of various figures. Derive three classes SQUARE, RECTANGLE, TRIANGLE from class Shape and Calculate Area and Perimeter of each class separately and display the result.

### **PROCEDURE:**

- Step 1 : Start the process.
- Step 2 : Include the header files iostream.h and conio.h.
- Step 3 : Create a class SHAPE with member variables l,b,area and peri as float. Step 4 : Using getdata() to get the input for the length and breath.
- Step 5 : Declare virtual function calc\_area() and calc\_peri() in class SHAPE.
- Step 6 : Create a derived class SQUARE, RECTANGLE and TRIANGLE from the base class SHAPE.
- Step 7 : Create a member function calc\_area() and calc\_peri() in all the derived class and calculate the area and perimeter of the class SQUARE, RECTANGLE and TRIANGLE.

## **DEPARTMENT OF COMPUTER APPLICATIONS**



## C++ PROGRAMMING LAB

- Step 8 : In main() function create an pointer object of the class SHAPE. Step 9 : Display the different shapes name with number.
- Step 10 : Get the input from the user for the shape with the variable opt.
- Step 11 : In switch case statement pass opt as argument and calculate area and perimeter according to the choice and display the output.
- Step 12 : Stop the process.

### SOURCE CODE:

```
#include<iostream.h>
#include<conio.h>
class SHAPE
public:
float l,b,area,peri; void getdata()
{
cout<<"\n\n Enter the values of length and breath:"; cin>>l>>b;
}
virtual void calc_area()
{
}
virtual void calc_peri()
{
}
};
class SQUARE:public SHAPE
{
int a; public:
void calc_area()
{
cout<<"\n Enter the value for side of square:"; cin>>a;
area=a*a;
cout<<"\n\n Area of square="<<area;
}
void calc_peri()
{
peri=4*a;
cout<<"\n\n Perimeter of square ="<<peri;
}
};

class RECTANGLE:public SHAPE

void calc_area()
{
getdata(); area=l*b;
cout<<"\n\n Area of rectangle="<<area;
}
```



## C++ PROGRAMMING LAB

```
void calc_peri()
{
    peri=2*(l+b);
    cout<<"\n\n Perimeter of rectangle="<<peri;
}
};
class TRIANGLE:public SHAPE
{
public:
    void calc_area()
    {
        getdata(); area=(l*b)/2.0;
        cout<<"\n\n Area of the triangle ="<<area;
    }
    void calc_peri()
    {
        int h;
        cout<<"\n\n Enter height="; cin>>h;
        peri=l+b+h;
        cout<<"\n\n Perimeter of triangle="<<peri;
    }
};

void main()
{
    int opt;

    SHAPE *ptr; clrscr();
    cout<<"\n\n\t\t 1.SQUARE \n\n\t\t 2.RECTANGLE \n\n\t\t 3.TRIANGLE ";
    cout<<"\n\n\t Enter your option:";
    cin>>opt; switch(opt)
    {
        case 1:
            SQUARE s;
            ptr=&s;
            ptr->calc_area(); ptr->calc_peri(); break;
        case 2:
            RECTANGLE r;
            ptr=&r;
            ptr->calc_area(); ptr->calc_peri(); break;
        case 3:
            TRIANGLE t;
            ptr=&t;
            ptr->calc_area(); ptr->calc_peri(); break;
        default:
            cout<<"\n\n\t\t INVALID OPTION"; break;
    }
    getch();
}
```



## **C++ PROGRAMMING LAB**

### **OUTPUT:**

1. SQUARE
2. RECTANGLE
3. TRIANGLE

Enter your option:1

Enter the value of side for square : 5 Area of Square = 25  
Perimeter of Square = 20

1. SQUARE
2. RECTANGLE
3. TRIANGLE

Enter your option:2

Enter the values of length and breath:2 3 Area of the rectangle = 6  
Perimeter of rectangle = 10

1. SQUARE
2. RECTANGLE
3. TRIANGLE

Enter your option:3

Enter the values of length and breath:2 3 Area of the triangle = 3  
Enter height = 5 Perimeter of triangle = 10

### **CONCLUSION:**

The above program has been verified and executed successfully.

\



## C++ PROGRAMMING LAB

### EXPERIMENT 8

#### AIM:

Write a C++ Program to create two classes each class consists of two private variables, a integer and a float variable. Write member functions to get and display them. Write a FRIEND Function common to both classes, which takes the object of above two classes as arguments and the integer and float values of both objects separately and display the result.

#### PROCEDURE:

- Step 1 : Start the process.
- Step 2 : Include the header file iostream.h and conio.h. Step 3 : Declare two classes namely class1 and class2.
- Step 4 : In each class declare private member variable with integer and float data type.
- Step 5 : Using the member function getdata() to get the input from the user for the integer and float variable.
- Step 6 : The display() member function in each class will display the value of integer and float variables.
- Step 7 : In both the classes declare the friend function result and display the both classes member variable values. Pass the argument as objects.
- Step 8 : In main function declare the objects for class1 and class2. Step 9 : Call the member function getdata() and display().
- Step 10 : Then call the function result to display the member variable's value of the both classes.
- Step 11 : Stop the process.

#### SOURCE CODE:

```
#include<iostream.h>
#include<conio.h> class class2;
class class1

{
    int a; float b; public:
    void getdata()
    {
        cout<<"\n Enter a and b values:"; cin>>a>>b;
    }
    void display()
    {
        cout<<"\n\n Class1 content \n\n"; cout<<"\n a="<<a<<"\n b="<<b;
    }
}

friend void result(class1,class2);

};

class class2
{
```





## C++ PROGRAMMING LAB

```
int x;
float y; public:
void getdata()
{
    cout<<"\n Enter x and y values:"; cin>>x>>y;
}
void display()
{
    cout<<"\n\n Class 2 content \n\n"; cout<<"\n x="<<x<<"\n y="<<y;
}

friend void result(class1,class2);

};

void result(class1 c1,class2 c2)

{

    cout<<"\n\n Class1 content using friend function\n\n"; cout<<"\n a="<<c1.a<<"\n b="<<c1.b;
    cout<<"\n \n Class2 content using friend function\n"; cout<<"\n x="<<c2.x<<"\n y="<<c2.y;
}

void main()

{

    clrscr();

    class1 cl1; class2 cl2; cl1.getdata(); cl2.getdata(); cl1.display(); cl2.display(); result(cl1,cl2); getch();
}
```

### OUTPUT:

```
Enter a and b values: 2 3.4
Enter x and y values: 5 6.7
Class1 content a=2
b=3.4
Class 2 content x=5
y=6.7
```

```
Class1 content using friend function a=2
b=3.4
```

```
Class2 content using friend function x=5
y=6.7
```



## **C++ PROGRAMMING LAB**

### **CONCLUSION:**

The above program has been verified and executed successfully.

### **EXPERIMENT 9**

#### **AIM:**

#### **MATRIX ADDITION USING FUNCTION OVERLOADING**

Aim:

Write a C++ Program using Function Overloading to read two Matrices of different Data Types such as integers and floating-point numbers. Find out the sum of the above two matrices separately and display the sum of these arrays individually.

#### **PROCEDURE:**

Algorithm:

Step 1 : Start the process.

Step 2 : Include the header files iostream.h and conio.h.

Step 3 : Declare the public variables I,j,c1,c2,r1,r2 and a[10][10] as integer, b[10][10] and c[10][10] as float.

Step 4 : In the main() function get the input from the user for number of columns and number of rows using the variable c1, r1 and c2, r2 for the matrix a and matrix b.

Step 5: The matrix addition is possible when both the matrix a and b are having same number of columns and rows.

Step 6 : If both rows and columns are same get input from user using get() function overloading for integer matrix a and float matrix b.

Step 7 : The function sum() is used to add matrix a and b, store the result into the matrix c.

Step 8 : The display() function is used to display the matrix c values in matrix format. Step 9 : Stop the process.

#### **SOURCE CODE:**

```
Source Code: #include<iostream.h> #include<conio.h>
int i,j,c1,c2,r1,r2,a[10][10];

float b[10][10],c[10][10];

void get(int &x)
```



## C++ PROGRAMMING LAB

```
{

cin>>x;

}

void get(float &x)

{

cin>>x;

}

void sum()

{

for(i=0;i<r1;i++) for(j=0;j<c1;j++)
c[i][j]=a[i][j]+b[i][j];

}

void display()

{

for(i=0;i<r1;i++)

{
for(j=0;j<c1;j++)
cout<<c[i][j]<<"\t"; cout<<"\n";
}
}

void main()
{
clrscr();
cout<<"\n\n Enter row,col for matrix1:"; cin>>r1>>c1;
cout<<"\n\n Enter row,col for matrix2:"; cin>>r2>>c2;
if((r1!=r2||c1!=c2))
{
cout<<"\n Matrix addition is not possible"; goto end;
}
cout<<"\n Enter matrix1\n\n"; for(i=0;i<r1;i++)
for(j=0;j<c1;j++) get(a[i][j]);
cout<<"\n Enter matrix2\n\n"; for(i=0;i<r2;i++)
for(j=0;j<c2;j++) get(b[i][j]);
sum();
}
```



## **C++ PROGRAMMING LAB**

```
cout<<"The Resultant matrix\n"; display();  
end: getch();  
}
```

### **OUTPUT:**

Enter row,col for matrix1: 2 2

Enter row,col for matrix2: 2 2 Enter matrix1

1  
2  
3  
4

Enter matrix2 1.1

2.2  
3.3  
4.4

The Resultant Matrix    2.1   4.2  
6.3     8.4

### **CONCLUSION:**

The above program has been verified and executed successfully.

## **EXPERIMENT 10**

### **AIM:**

Write a C++ Program to check whether the given string is a palindrome or not using Pointers.

### **PROCEDURE:**

Step 1 : Start the process.

Step 2 : Include the header files iostream.h, string.h, and conio.h.

Step 3 : In main() function declare the str1[20], str2[20], \*s and \*rs as character data type.

Step 4 : Get the input from the user for the variable str1. Step 5 : Using the strcpy() function copy str1 into str2. Step 6 : Assign the str1 to pointer variable s.

Step 7 : With the help of function strev(), reverse the string of str2 and assign to pointer variable rs.

Step 8 : Compare two pointers s and rs using the function strcmp(), if it returns 0 print the given string is palindrome. Otherwise print the given string is not palindrome.

Step 9 : Stop the process.

## **DEPARTMENT OF COMPUTER APPLICATIONS**



## **C++ PROGRAMMING LAB**

### **SOURCE CODE:**

```
#include<iostream.h>
#include<string.h>
#include<conio.h>
void main()
{
    char str1[20],str2[20],*s,*rs;
    clrscr();
    cout<<"\n\n Enter a string:";
    cin>>str1;
    strcpy(str2,str1);
    s=str1;
    rs=strrev(str2);
    if(strcmp(s,rs)==0)
    cout<<"\n\n The given string is palindrome";
    else
    cout<<"\n\n The given string is not a palindrome"; getch();
}
```

### **OUTPUT:**

```
Enter a string : mom
The given string is palindrome
Enter a string : book
The given string is not palindrome
```

### **CONCLUSION:**

The above program has been verified and executed successfully.

## **EXPERIMENT 11**

### **AIM:**

Write a C++ Program to create a File and to display the contents of that file with line numbers.

### **PROCEDURE:**

- Step 1 : Start the process.
- Step 2 : Create an empty file named as "line.txt".
- Step 3 : Include the header files iostream.h, conio.h and fstream.h.

## **DEPARTMENT OF COMPUTER APPLICATIONS**



## **C++ PROGRAMMING LAB**

- Step 4 : Declare object f from fstream class, c as character and lno as integer variable.  
Step 5 : Open a file line.txt with the help of the member function open from the object f.  
Step 6 : Write the few lines in the file line.txt using insertion operator.  
Step 7 : Using the seekg() function set the file pointer to the beginning of the file.  
Step 8 : With while() loop read each line of the file and display the content with line number.  
Step 9 : Close the file line.txt. Step 10 : Stop the process.

### **SOURCE CODE:**

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
void main()
{

    fstream f; char c; int lno=1;
    f.open("line.txt",ios::in|ios::out|ios::trunc);
    f<<"C \n C++ \n JAVA \n COBOL \n PASCAL";
    f.seekg(0);
    clrscr();
    cout<<lno++<<" . ";
    while(f)
    {
        f.get(c);
        if(c=="\n") cout<<"\n"<<lno++<<" . ";
        else
            cout<<c;
    }

    f.close();
    getch();
}
```

### **OUTPUT:**

1. C
2. C++
3.     JAVA
4.     COBOL
5.     PASCAL

### **CONCLUSION:**

The above program has been verified and executed successfully.



## **C++ PROGRAMMING LAB**

### **EXPERIMENT 12**

#### **AIM:**

Program 12:

MERGE TWO FILES

Aim:

Write a C++ Program to merge two files into a single file.

#### **PROCEDURE:**

Step 1 : Start the process.

Step 2 : Create two files namely file1.txt and file2.txt with different content.

Step 3 : Include the header files fstream.h, conio.h and iostream.h.

Step 4 : Declare an object in and out from ifstream and ofstream.

Step 5 : Open a file file1.txt in write mode and file2.txt in read mode.

Step 6 : Using while loop read the content of file2.txt and write into the file file1.txt. Now file2.txt content are merged with file1.txt.

Step 7 : Close both the files file1.txt and file2.txt.

Step 8 : Stop the process.

#### **SOURCE CODE**

```
#include<fstream.h>
#include<conio.h>
#include<iostream.h>
void main()
{

ifstream in; ofstream out; char c;

out.open("file1.txt",ios::ate);

in.open("file2.txt",ios::in); while(in)
{

in.get(c); // Read from file2 out.put(c); // Write into file1
}

in.close();

out.close();

}
```



## **C++ PROGRAMMING LAB**

### **OUTPUT:**

C:\TurboC++>Type File1.txt

Computing Fundamentals and C Programming C:\TurboC++>Type File2.txt  
C++ Programming

After Program Execution:

C:\TurboC++>Type File1.txt

Computing Fundamentals and C Programming C++ Programming

### **CONCLUSION:**

The above program has been verified and executed successfully.





## **C++ PROGRAMMING LAB**