ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ, ಬೆಳಗಾವಿ

VISVESVARAYA TECHNOLOGICAL UNIVERSITY - BELAGAVI

Department of Computer Science and Engineering

"Jnana Sangama", VTU-Campus, Belagavi-590018

LAB-JOURNAL

Academic Year: 2025-26

Name :

USN :

Sem : 3rd Semester

Subject Name : Object Oriented Programming with C++

Subject Code : BCS306B

Course : Bachelor of Technology

Programme : Computer Science and Engineering

CERTIFICATE

This is certify that Mr./Mrs._____with

USN _____has satisfactorily completed all the Laboratory

Assignment of Subject Object-Oriented Programming with C++ having

Subject Code BCS306B during the academic year 2025-26.

Faculty in-charge **Programmer Co-ordinator**

Signature of the Examiners

INDEX

Program NO.	DATE	NAME OF THE PROGRAM	PAGE NO.	SIGN
1		Develop a C++ program to find the largest of three numbers.		
2		Develop a C++ program to sort the elements in ascending and descending Order.		
3		Develop a C++ program using classes to display student name, roll number, marks obtained in two subjects and total score of the student.		
4		Develop a C++ program for a bank empolyee to print name of the employee, account_no. & balance. Print invalid balance if amount<500, Display the same, also display the balance after withdraw and deposit.		
5		Develop a C++ program to demonstrate function overloading for the following prototypes. add(int a, int b) add(double a, double b).		
6		Develop a C++ program using Operator Overloading for overloading Unary minus operator.		
7		Develop a C++ program to implement Multiple inheritance for performing arithmetic operation of two numbers.		
8		Develop a C++ program using Constructor in Derived classes to initialize alpha, beta and gamma and display corresponding values.		
9		Develop a C++ program to create a text file, check file created or not, if created it will write some text into the file and then read the text from the file.		
10		Develop a C++ program to create a text file, check file created or not, if created it will write some text into the file and then read the text from the file.		
11		Develop a function which throws a division by zero exception and catch it in catch block. Write a C++ program to demonstrate usage of try, catch and throw to handle exception.		
12		Develop a C++ program that handles array out of bounds exception using C++.		

1. Develop a C++ program to find the largest of three numbers

```
#include<iostream>
using namespace std;
 int main() {
double n1, n2, n3;
cout << "Enter three numbers: "; cin
>> n1 >> n2 > n3;
if(n1 >= n2 && n1 >= n3)
cout << "Largest number is : " << n1; else
if (n2 >= n1 && n2 >= n3)
cout << "Largest number is: " << n2;
else
cout << "Largest number is : " << n3;
return 0;
}
```

Output:

```
Enter three numbers: 10
20
15
Largest number is: 20
```

2. Develop a C++ program to sort the elements in ascending and descending order

```
#include <iostream>
using namespace std;

int main(){
int num[100],n;
int i,j,temp;
cout<<"Enter n for the numbers you want to sort"<<"\n";
cin>>n;
for(i=0;i<n;i++){
cout<<"Enter numbers:"<<"\n";
cin>>num[i];}
for(i=0;i<n;i++){
for(j=i+1 ;j<n;j++){
if(num[i]> num[j])
{
temp=num[i];
num[i]=num[j];
num[j]=temp;
}
}}
cout<<"Ascending order :"<<"\n";
for(i=0;i<n;i++)
{
cout<<" "<<num[i] <<"\n";
}
for(i=0;i<n;i++){
for(j=i+1;j<n;j++){
if(num[i]<num[j]){
temp=num[i]; num[i]=num[j];
num[j]=temp;}}}
cout<<" Descending order : "<<"\n";
        for(i=0;i<n;i++)
```

```
{
cout<<" "<<num[i] <<"\n";

    }
return 0;
}
```

Output

```
Enter n for the numbers you want to sort
5
Enter numbers
10
21
13
14
11
Ascending order:
10
11
13
14
21
Descending order :
21
14
13
11
10
```

3. **Develop a C++ program using classes to display student name, roll number, marks obtained in two subjects and total score of student**

```
#include <iostream>
#include <conio.h>
#include <string.h>
using namespace std;


class Student {
private:
 char name[50];
 int rollNumber;
 int marksSubject1;
 int marksSubject2;

public:
    void setDetails(char* studentName, int roll, int marks1, int marks2) {
    strcpy(name, studentName);
    rollNumber = roll;
    marksSubject1 = marks1;
    marksSubject2 = marks2;
  }


  void displayDetails() {
    clrscr(); // Clear the screen (Turbo C++ specific)
    cout << "Student Name: " << name << endl;
    cout << "Roll Number: " << rollNumber << endl;
    cout << "Marks in Subject 1: " << marksSubject1 << endl;
    cout << "Marks in Subject 2: " << marksSubject2 << endl;
    cout << "Total Score: " << (marksSubject1 + marksSubject2) << endl;
  }
};
```

```
int main() {
    Student student;
    char name[50];
    int roll, marks1, marks2;
    cin>>name;
    cout << "Enter roll number: ";
    cin >> roll;
    cout << "Enter marks obtained in subject 1: ";
    cin >> marks1;
    cout << "Enter marks obtained in subject 2: ";
    cin >> marks2;
    student.setDetails(name, roll, marks1, marks2);
    cout << endl;
```

Output:

Enter student name: Samuel
Enter roll number: 0123
Enter marks obtained in subject 1: 35
Enter marks obtained in subject 2: 45
Student Name: Samuel
Roll Number: 0123
Marks in Subject 1: 35
Marks in Subject 2: 45
Total Score: 80

4. **Develop a C++ program for a bank employee to print name of the employee, account_no. & balance. Print invalid balance if amount<500, Display the same, also display the balance after withdraw and deposit.**

```
#include<iostream>
#include<conio.h>
#include<string.h>
using namespace std;
class BankEmployee{
private:
char name[50];
long accountNumber;
float balance;
public:
BankEmployee(const char ename[],long accno,float inibal)
{

strcpy(name,ename);
accountNumber=accno;
if(inibal<500){
cout<<"Invalid balance enter initial balance above 500"<<endl;
cin>>balance
}

else{
balance=inibal;
}

}

void displayDetails(){
cout<<"Employee Name: "<<name<<endl;
cout<<"Account Number: "<<accountNumber<<endl;
cout<<"Balance: "<<balance<<endl;
}

void withdraw(float amount){
if (balance >=amount){
```

```
balance - =amount;
cout<<"Balance after withdrawal: "<<balance<<endl;
}
else{
cout<<"Insufficient Balance"<<endl;
}
}
void deposit(float amount){
balance +=amount;
cout<<"Balance after deposit:"<<balance<<endl;
}
};
int main(){
char name[50];
long accno;
float inibal, wamount, damount;
clrscr();
cout<<"Enter Employee Name:";
cin.getline(name, sizeof(name));
cout<<"Enter Account Number:";
cin>>accnor;
cout<<"Enter initial balance:";
cin>>inibal;
BankEmployee employee(name,accountNumber, inibal);
cout<<endl;
cout<<"Employee Details:"<<endl;
employee.displayDetails();
cout<<"Enter the amount to withdraw: ";
cin>>wamount;
employee.withdraw(wamount);
cout<<"Enter the amount to deposit: ";
cin>>damount;
employee.deposit(damount);
```

```
getch();
return 0;
}
```

Output:

Enter Employee Name: Rakesh S
Enter Account Number: 2010268
Enter initial balance: 1000

Employee Details:
Employee Name: Rakesh S

Account Number: 2010268
Balance : 1000

Enter the amount to withdraw: 300
Balance after withdrawal: 700

Enter amount to deposit : 200
Balance after deposit:900

5. **Develop a C++ program to demonstrate function overloading for the following prototypes. add(int a, int b) add(double a, double b)**

```
#include<iostream.h>
#include<conio.h>
int add(int a,int b)
{
return a+b;
}

double add(double a,double b)
{
return a+b;
}

int main()
{
clrscr();
int intResult=add(5,3);
double doubleResult=add(2.5,3.7);
cout<<"Result of adding two integers: "<<intResult<<endl;
cout<<"Result of adding two double: "<<doubleResult<<endl;
getch();
return 0;
}
```

Output:

Result of adding two integers: 8
Result of adding two double: 6.2

6. **Develop a C++ program using Operator Overloading for overloading Unary minus operator**

```
#include<iostream.h>
#include<conio.h>
class Number{
private:
        int value;
public:
        Number():value(0){}
        Number(int val):value(val){}
        Number operator-() const{
                Number result(-value);
                return result;
        }
        void display() const{
                cout<<"Value: "<<value<<endl;
        }
};
int main(){
clrscr();
        Number num1(10);
        Number num2=-num1;
        cout<<"Original number: ";
        num1.display();
        cout<<"After Unary Minus: ";
        num2.display();
        getch();
        return 0;
}
```

Output:

Original number: Value : 10
After Unary Minus: Value : -10

7. **Develop a C++ program to implement Multiple inheritance for performing arithmetic operation of two numbers**

```
#include<iostream.h>
#include<conio.h>
class addition
{
public:
int add(int a,int b){
                return a+b;
        }
};
class subtraction{

public: int subtract(int a,int b){
                return a-b;
        }
};

class arithmetic:public addition,public substraction
{
public: void display(int a,int b){

        cout<<"Sum= "<<add(a,b)<<endl;
        cout<<"Difference= "<<substract(a,b)<<endl;
        }
};

int main()
{
clrscr();
int num1,num2;
```

```
 cout<<"Enter two number: "<<endl;
 cin>>num1>>num2;
 arithmetic obj;
obj.display(num1,num2);
getch();
 return 0;
 }
```

Output:

```
Enter two number: 10  20
Sum= 30
Difference = -10
```

8. **Develop a C++ program using Constructor in Derived classes to initialize alpha, beta and gamma and display corresponding values.**

```
#include<iostream.h>
#include<conio.h>
class Alpha{
protected: int alpha;
public: Alpha(int a):alpha(a){}
};

class Beta:public Alpha{
protected: int beta;
public: Beta(int a,int b):Alpha(a),beta(b){}
};

class Gamma:public Beta
{

private: int gamma;
public: Gamma(int a,int b,int c):Beta(a,b),gamma(c)
{}

        void displayvalues(){
        cout<<"Alpha: "<<alpha<<endl;
        cout<<"Beta: "<<beta<<endl;
        cout<<"Gamma: "<<gamma<<endl;
        }

};

int main(){
clrscr();
Gamma obj(10,20,30);
obj.displayvalues();
getch();
return 0;}
```

Output:

Alpha: 10
Beta: 20
Gamma:30

9. Develop a C++ program to create a text file, check file created or not, if created it will write some text into the file and then read the text from the file.

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
int main(){
        clrscr();
        ofstream outFile("sample.txt");
        if(!outFile){
                cout<<"Error creating the file!"<<endl;
                getch();
                return 1;
        }
        outFile<<"Hello this is some text written to the file.\n";
        outFile<<"This is another line in the file.";
        outFile.close();
        ifstream inFile("sample.txt");
        if(!inFile){
                cout<<"Error opening the file!"<<endl;
                getch();
                return 1;
        }
        cout<<"Contents of the file:"<<endl;
        char ch;
        while(inFile.get(ch)){
                cout<<ch;
        }
        inFile.close();
        getch();
        return 0;}
```

Output:

Contents of the file:
Hello this is some text written to the file
This is another line in the file.

10. Develop a C++ program to write and read time in/from binary file using fstream.

```
#include <iostream.h>
#include <fstream.h>
#include <iomanip.h> //for setfill() and setw()


#define FILE_NAME "time.dat"

void writeTime(int h, int m, int s){

        char str[10];

        fstream file;
        file.open(FILE_NAME, ios::out|ios::binary);

        if(!file){
                cout<<"Error in creating file!!!"<<endl;
                return;
        }

        //make string to write
        sprintf(str,"%02d:%02d:%02d",h,m,s);

        //write into file
        file.write(str,sizeof(str));
        cout<<"Time "<<str<<" has been written into file."<<endl;

        //close the file
        file.close();

}
```

```
void readTime(int *h,int *m, int *s){

	char str[10];
	int inH,inM,inS;

	fstream finC;
	finC.open(FILE_NAME,ios::in|ios::binary);
	if(!finC){
		cout<<"Error in file opening..."<<endl;
		return;
	}
	if(finC.read((char*)str,sizeof(str))){
		//extract time values from the file
		sscanf(str,"%02d:%02d:%02d",&inH,&inM,&inS);
		//assign time into variables, which are passing in function
		*h=inH;
		*m=inM;
		*s=inS;
	}
	finC.close();
}

int main(){
	int m,h,s;

	cout<<"Enter time:\n";
	cout<<"Enter hour: ";	cin>>h;
	cout<<"Enter minute: "; cin>>m;
	cout<<"Enter second: "; cin>>s;

	//write time into file
	writeTime(h,m,s);

	//now, reset the variables
```

```
        h=m=s=0;
       readTime(&h,&m,&s);
      cout<<"The time is
"<<setw(2)<<setfill('0')<<h<<":"<<setw(2)<<setfill('0')<<m<<":"<<setw(2)<<setfill('0')
<<s<<endl;

        return 0;
}
```

Output:

```
Enter time:
Enter hour: 10
Enter minute: 22
Enter second: 45
Time 10:22:45 has been written into file.
The time is 10:22:45
```

11. Develop a function which throws a division by zero exception and catch it in the catch block. Write a C++ program to demonstrate usage of try , catch and throw to handle exception.

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

int main() {
    int numerator, denominator, result;
    cout << "Enter First Number: ";
    cin >> numerator;
    cout << "Enter Second Number: ";
    cin >> denominator;
    try {
        if (denominator == 0) {
            throw runtime_error("Division by zero error");
        }

        result = numerator / denominator;
        cout << "Result: " << result << endl;
    } catch (const runtime_error& ex) {
        cerr << "Exception caught: " << ex.what() << endl;
        return 1; // Return an error code, if desired
    }

    return 0;
}
```

Output:

Enter First Number: 20
Enter Second Number:0
Exception caught:
Division by zero error

12. Develop a C++ program that handles Array Out of Bounds exception using C++

```
include <iostream.h> int
main() {
 const int arraySize = 5; int
 myArray[arraySize];

 try {
    int indexOutOfRange = myArray[arraySize]; cout
    << "This line won't be executed." << endl;
 } catch (...) {
    cerr << "Exception caught: Array index out of bounds." << endl;
 }


 cout << "Program continues after exception handling." << endl;

 return 0;
}
```

Output:

Exception caught: Array index out of bounds. Program
continues after exception handling.