ACHARYA INSTITUTE OF TECHNOLOGY
Acharya Dr. Sarvepalli Radhakrishnan Road, Bangalore-560 107
DEPARTMENT OF Information Science & Engineering

Introduction to C++ Programming (22PLC15D/22PLC25D)

Programming Assignments

I/II Semester
(Common to all branches)
LAB MANUAL

Complied and Executed By:
Pankaj Kumar
Assistant Professor
Dept of ISE

The versioned repository of all the programs can be found here as a Github Repository
https://github.com/mepankaj/Introduction-to-C-Programming-Lab-22PLC15D-25D-

How to Run C++ program:

Method 1:

Using Turbo C++ for Windows with full window screen mode to compile and execute a C++ Program

Video Demo : https://www.youtube.com/watch?v=9PRiad4R6IY

Method 2:

Download and install Dev C++ Software to run C++ Program

Video Demo: https://www.youtube.com/watch?v=J9fkOicyad8

List of Program

1. Write a C++ program to sort the elements in ascending and descending order.

2. Write a C++ program to find the sum of all the natural numbers from 1 to n.

3. Write a C++ program to swap 2 values by writing a function that uses call by reference technique.

4. Write a C++ program to demonstrate function overloading for the following prototypes.

add(int a, int b)

add(double a, double b)

5. Create a class named Shape with a function that prints "This is a shape". Create another class named Polygon inheriting the Shape class with the same function that prints "Polygon is a shape". Create two other classes named Rectangle and Triangle having the same function which prints "Rectangle is a polygon" and "Triangle is a polygon" respectively. Again, make another class named Square having the same function which prints "Square is a rectangle".Now, try calling the function by the object of each of these classes.

6.Suppose we have three classes Vehicle, FourWheeler, and Car. The class Vehicle is the base class, the class FourWheeler is derived from it and the class Car is derived from the class FourWheeler. Class Vehicle has a method 'vehicle' that prints 'I am a vehicle', class FourWheeler has a method 'fourWheeler' that prints 'I have four wheels', and class Car has a method 'car' that prints 'I am a car'. So, as this is a multi-level inheritance; we can have access to all the other classes methods from the object of the class Car. We invoke all the methods from a Car object and print the corresponding outputs of the methods.

So, if we invoke the methods in this order, car(), fourWheeler(), and vehicle(), then the output will be

I am a car

I have four wheels

I am a vehicle

Write a C++ program to demonstrate multilevel inheritance using this.

7. Write 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.

8.Write aC++ program to write and read time in/from binary file using fstream

9. Write 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.

10. Write a C++ program function which handles array of bounds exception using C++.

1. Write a C++ program to sort the elements in ascending and descending order.

```
#include<iostream>
using namespace std;
int main ()
{
   int num[20],n;
   int i, j, temp;
   cout<<"\n Enter the size of Array: \n";
   cin>>n;
   cout<<"Enter the Array value";
   for (i = 0; i < n; ++i)
   cin>>num[i];
  for (i = 0; i < n; ++i)   //  'for' loop is used for sorting the numbers in descending order
   {
      for (j = i + 1; j < n; ++j)
      {
         if (num[i] < num[j])
         {
            temp = num[i];
            num[i] = num[j];
            num[j] = temp;
         }
      }
   }
   cout<<"\n Numbers in Descending Order : \n";
   for (i = 0; i < n; ++i)
   {
      cout<<" ";
      cout<<num[i];
      cout<<"\n";
   }
for(i=0;i<n;i++)   //outer-loop for  sorting the numbers in ascending order
{
  for(int j=0;j<n;j++)  //inner-loop
   {
     if(num[i]<num[j]) // represent second element in the array list
```

```
      {
         temp = num[i];    // first array element assign to variable temp
         num[i] = num[j]; // second element assigning to first element
         num[j] = temp; // variable temp (means first element) assigning to second element
      }
    }
}
cout<<"\n Numbers in Ascending Order : \n";
   for (i = 0; i < n; ++i)
   {
      cout<<" ";
      cout<<num[i];
      cout<<"\n";
   }
}
```

Output:

```
Enter the size of Array:
10
Enter the Array value
8 10 6 9 3 5 2 4 1 7
Numbers in Descending Order :
 10  9  8  7  6  5  4  3  2  1
 Numbers in Ascending Order :
 1  2  3  4  5  6  7  8  9  10
```

2. Write a C++ program to find the sum of all the natural numbers from 1 to n.

Program:

```
#include<iostream>
using namespace std;
int main()
{
   int n;
   cout << "Enter a number : ";
   cin >> n;
      int sum=0;
      for(int i=1;i<=n;i++)
      sum+=i;
    cout <<"Sum of " <<n <<" natural numbers is:\n"<< sum;
   return 0;
}
```

Output:

```
Enter a number : 50
Sum of 50 natural numbers is:
1275
```

3. Write a C++ program to swap 2 values by writing a function that uses call by reference technique.

Program:

```
#include<iostream>
using namespace std;
void swap(int &,int &);
int main()
{
	int x,y;
	cout<<"\nEnter 1st number :: ";
	cin>>x;
	cout<<"\nEnter 2nd number :: ";
	cin>>y;
	cout<<"\n Before swapping the numbers are:"<<"\n\t x = "<<x<<"\n\t y = "<<y<<endl;
	swap(x,y);
	cout<<"\n After swapping the numbers are:"<<"\n\t x = "<<x<<"\n\t y = "<<y<<endl;
	return 0;
}

void swap (int &num1, int &num2) //&num1 and &bnum2 are Reference variables
{
	int temp;
	temp=num1;
	num1=num2;
	num2=temp;
}
```

Output:

```
Enter 1st number :: 10
Enter 2nd number :: 5
Before swapping the numbers are:
      X = 10
      y = 5

 After swapping the numbers are:
      X = 5
      y = 10
```

4. **Write a C++ program to demonstrate function overloading for the following prototypes.**

add(int a, int b)

add(double a, double b)

Program:

```
#include <iostream>
using namespace std;
// function with 2 integer parameters
void add(int a, int b) {
   cout << "Integer number1: " << a<< endl;
   cout << " and Integer number2: " << b << endl;
}
// function with 2 double parameter
void add(double p, double q) {
   cout << "Double number1: " << p << endl;
   cout << "Double number2: " << q << endl;
}
int main() {
   int x = 5 , y=10;
   double l = 5.5, m=10.5;
   // call function with int  parameter
   add(x,y);
   // call function with double parameters
   add(l, m);
   return 0;
}
```

Output:

```
Demo of Function Overloading
Integer number1: 5
 and Integer number2: 10
Double number1: 5.5
 and Double number2: 10.5
```

5. Create a class named Shape with a function that prints "This is a shape". Create another class named Polygon inheriting the Shape class with the same function that prints "Polygon is a shape". Create two other classes named Rectangle and Triangle having the same function which prints "Rectangle is a polygon" and "Triangle is a polygon" respectively. Again, make another class named Square having the same function which prints "Square is a rectangle".Now, try calling the function by the object of each of these classes.

Program:

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

class Shape{
   public:
   void show();
};
void Shape::show(){
   cout << "This is a Shape" << endl;
}

class Polygon : public Shape{
   public:
```

```
    void show();
};
void Polygon::show(){
    cout << "Polygon is a Shape" << endl;
}

class Triangle : public Polygon{
    public:
    void show();
};
void Triangle::show(){
    cout << "Triangle is a Polygon" << endl;
}

class Rectangle : public Polygon{
    public:
    void show();
};
void Rectangle::show(){
    cout << "Rectangle is a Polygon" << endl;
}

class Square : public Rectangle{
    public:
    void show();
};
void Square::show(){
    cout << "Square is a Rectangle" << endl;
}
```

Output:

```
This is a Shape
Polygon is a Shape
Rectangle is a Polygon
Triangle is a Polygon
Square is a Rectangle
```

6.Suppose we have three classes Vehicle, FourWheeler, and Car. The class Vehicle is the base class, the class FourWheeler is derived from it and the class Car is derived from the class FourWheeler. Class Vehicle has a method 'vehicle' that prints 'I am a vehicle', class FourWheeler has a method 'fourWheeler' that prints 'I have four wheels', and class Car has a method 'car' that prints 'I am a car'. So, as this is a multi-level inheritance; we can have access to all the other classes methods from the object of the class Car. We invoke all the methods from a Car object and print the corresponding outputs of the methods.

So, if we invoke the methods in this order, car(), fourWheeler(), and vehicle(), then the output will be

I am a car

I have four wheels

I am a vehicle

Write a C++ program to demonstrate multilevel inheritance using this.

Program:

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

class Vehicle{
```

```
    public:
    void vehicle();
};
void Vehicle::vehicle(){
    cout << "I am a vehicle" << endl;
}

class FourWheeler : public Vehicle{
    public:
    void fourWheeler();
};
void FourWheeler::fourWheeler(){
    cout << "I have four wheels" << endl;
}

class Car : public FourWheeler{
    public:
    void car();
};
void Car::car(){
    cout << "I am a car" << endl;
}
int main() {
    Car myCar;

    myCar.car();
    myCar.fourWheeler();
    myCar.vehicle();
    return 0;
}
```

Output:

```
I am a car
I have four wheels
I am a vehicle
```


7. Write 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.

Program:

```
#include <iostream>
#include <fstream>
using namespace std;
int main() {

   string flName;
   char mesg[40], ch;

   cout << "Enter the file name you want to create : ";
   cin >> flName;
   cin.get();      //read the trailing enter character
```

```
    ofstream fout(flName.c_str());
    // fout.close();
    if(fout.fail()){
        cout << "\nFailed to create file." << endl;
    }
    else{
        cout << "\nFile " << flName <<" created successfully" << endl;
    }

    cout << "Enter a message : ";
    cin.getline(mesg,40);
    fout << mesg << endl;
    cout << "\nMessage written to file successfully\n" << endl;
    fout.close();

    ifstream fin(flName.c_str());
    cout << "Here are the contents of " << flName << ":\n";

    while (fin.get(ch))    // read character from file and
        cout << ch;    // write it to screen
    cout << "\nDone reading file contents\n" << endl;
    fin.close();
    return 0;
}
```

Output:

```
Enter the file name you want to create : random.txt
```

```
File random.txt created successfully
 Enter a message : The universe is vast and endless
Message written to file successfully
 Here are the contents of random.txt:
The universe is vast and endless
Done reading file contents
```

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

Program:

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cstring>
using namespace std;
class timeVal{
   int hh, mm, ss;
  char ampm[3];
   public:
   void setdata(int h, int m, int s, const char* half)
   {
      hh = h;
      mm = m;
      ss = s;
```

```
        strcpy(ampm, half);
    }
    void showdata()
    {
        cout << "\nThe Time is : ";
        cout << setfill('0') << setw(2) << hh << ":";
        cout << setfill('0') << setw(2) << mm << ":";
        cout << setfill('0') << setw(2) << ss << " ";
        cout << ampm << endl << endl;

    }
};

int main()
{
    timeVal writeObj, readObj;
    int hh, mm, ss;
    char ampm[3];
    cout << "Enter Hours : "; cin >> hh;
    cout << "Enter Minutes : "; cin >> mm;
    cout << "Enter Seconds : "; cin >> ss;
    cout << "Enter am or pm : "; cin >> ampm;
    writeObj.setdata(hh,mm,ss,ampm);
    ofstream outFile("TimeFile", ios::out | ios::binary);
    if(!outFile) {
        cout << "Cannot open file.\n";
        return 1;
    }
    outFile.write((char *) &writeObj, sizeof(timeVal));
    cout << "\nWritten the time object successfully to binary file" << endl;
```

```
    outFile.close();

    // now, read back;
    ifstream inFile("TimeFile", ios::in | ios::binary);
    if(!inFile) {
       cout << "Cannot open file.\n";
       return 1;
    }
    inFile.read((char *) &readObj, sizeof(timeVal));
    cout << "\nRead the time object successfully from binary file" << endl;
    readObj.showdata();
    inFile.close();
    return 0;
}
```

Output:

```
Enter Hours : 6
Enter Minutes : 35
Enter Seconds : 6
Enter am or pm : am

 Written the object successfully to binary file

Read the object successfully from binary file

The Time is : 06:35:06 am


Enter Hours : 8
Enter Minutes : 5
 Enter Seconds : 9
Enter am or pm : am
Written the object successfully to binary file

Read the object successfully from binary file
The Time is : 08:05:09 am
```

9. Write 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.

Program:

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

void fnDivide(int, int);

int main(void){

   int iNum1, iNum2;
```

```
    cout << "Enter the value of m and n : " ;
    cin >> iNum1 >> iNum2;

    try{
        fnDivide(iNum1, iNum2);
    }
    catch (logic_error& e){
        cout << "Processing error " << endl << e.what() << " occured.\n";
    }

    return 0;
}

void fnDivide(int v1, int v2){
    double dRes;
    if(v2 == 0)
        throw logic_error("Division by Zero Exception");
    dRes = (double)(v1)/v2;

    cout << v1 << " divided by " << v2 << " is equal to " << dRes << endl << endl;
}
```

Output:

```
Enter the value of m and n : 9 4
9 divided by 4 is equal to 2.25

Enter the value of m and n : 5 0
 Processing error
Division by Zero Exception occured.
```

10. Write a C++ program function which handles array of bounds exception using C++.

Program:

```
#include <iostream>
#include <iomanip>
#include <cstdlib>
#include <vector>
using namespace std;

int main(void){
```

```
    vector<int> values;
    int iDx, iElem;

    srand(time(NULL));
    for(int i=0;i<100;i++){
        iElem = rand() %10000;
        values.push_back(iElem);
    }

    try{
        //generate a random index and test repeatedly 10 times
        for(int i=0;i<10;i++){
            iDx = rand() % 200 - 50;
            if(iDx < 0 || iDx >=100){
                cout << "Generated index " << iDx << " is invalid" << endl;
                throw logic_error("Array out of Bounds");
            }
            cout << "Generated index is " << iDx << " the value at that index is : " <<
values[i] << endl;
        }

    }
    catch (logic_error& e){
        cout << "Processing error " << endl << e.what() << " exception occured.\n"
<< endl;
    }

    return 0;
}
```

Output:

```
Generated index is 37 the value at that index is : 8497
Generated index is 65 the value at that index is : 1027
Generated index is 7 the value at that index is : 8655
Generated index is 3 the value at that index is : 994
Generated index 105 is invalid
 Processing error
Array out of Bounds exception occured.

Generated index -31 is invalid
Processing error
Array out of Bounds exception occured.


Generated index is 43 the value at that index is : 8629
Generated index is 95 the value at that index is : 4852
Generated index is 45 the value at that index is : 9097
Generated index -9 is invalid
Processing error
Array out of Bounds exception occured.
```