assignment - 5

1. Write a C++ Program to Display Date using Constructors.

2. C++ program to Display Student Details using constructor and destructor.

3. C++ Program for Constructor with Parameters (Parameterized Constructor).

4. C++ Program to calculate Volume of Box using Constructor.

answer - 1

/\* C++ Program to Display Date using Constructors \*/

#include <iostream>

using namespace std;

class date

{

private:

int dd, mm, yy;

public:

date()

{

dd = 31;

mm = 12;

yy = 2016;

cout << "\nDate Object has been created..............\n";

}

void display()

{

cout << "\nThe Entered Date is :: ";

cout << dd << "-" << mm << "-" << yy << "\n";

}

};

int main()

{

date date1;

date1.display();

return 0;

}

answer - 2

/\* C++ program to Display Student Details using constructor and destructor \*/

#include<iostream>

using namespace std;

class stu

{

private:

char name[20],add[20];

int roll,zip;

public:

stu ();//Constructor

~stu();//Destructor

void read();

void disp();

};

stu :: stu()

{

cout<<"\nThis is Student Details constructor called..........."<<endl;

}

void stu :: read()

{

cout<<"\nEnter the student Name :: ";

cin>>name;

cout<<"\nEnter the student roll no :: ";

cin>>roll;

cout<<"\nEnter the student address :: ";

cin>>add;

cout<<"\nEnter the Zipcode :: ";

cin>>zip;

}

void stu :: disp()

{

cout<<"\nThe Entered Student Details are shown below ::---------- \n";

cout<<"\nStudent Name :: "<<name<<endl;

cout<<"\nRoll no is :: "<<roll<<endl;

cout<<"\nAddress is :: "<<add<<endl;

cout<<"\nZipcode is :: "<<zip;

}

stu :: ~stu()

{

cout<<"\n\nStudent Detail is Closed.............\n";

}

int main()

{

stu s;

s.read ();

s.disp ();

return 0;

}

answer - 3

/\*C++ Program for Constructor with Parameters (Parameterized Constructor).\*/

#include<iostream>

#include<conio.h>

using namespace std;

class Example {

// Variable Declaration

int a, b;

public:

//Constructor

Example(int x, int y) {

// Assign Values In Constructor

a = x;

b = y;

cout << "Im Constructor\n";

}

void Display() {

cout << "Values :" << a << "\t" << b;

}

};

int main() {

Example Object(10, 20);

// Constructor invoked.

Object.Display();

// Wait For Output Screen

getch();

return 0;

}

answer - 4

/\* C++ Program to calculate Volume of Box using Constructor.\*/

#include <iostream>

using namespace std;

class box

{

double length,width,height;

double volume;

public:

box(double a,double b,double c);

void vol();

};

box::box(double a,double b,double c)

{

length=a;

width=b;

height=c;

volume=length\*width\*height;

}

void box::vol()

{

cout<<"\nDimensions of Box are - \n";

cout<<"\nLength of Box :: "<<length<<"\n";

cout<<"\nWidth of Box :: "<<width<<"\n";

cout<<"\nHeight of Box :: "<<height<<"\n";

cout<<"\nVolume of Box :: "<<volume<<"\n";

}

int main()

{

box x(2.4,5.7,2.1),y(3.3,4.4,5.5);

x.vol();

y.vol();

return 0;

}

Assignment 6

1. Write a C++ program to read and print

employee

information with department and pf

information using

hierarchical inheritance.

Code-

 #include <iostream>

#include <stdio.h>

using namespace std;

//Base Class - basicInfo

class basicInfo {

protected:

    char name[30];

    int empId;

    char gender;

public:

    void getBasicInfo(void)

    {

        cout << "Enter Name: ";

        cin.ignore(1);

        cin.getline(name, 30);

        cout << "Enter Emp. Id: ";

        cin >> empId;

        cout << "Enter Gender: ";

        cin >> gender;

    }

};

//Base Class - deptInfo

class deptInfo : private basicInfo {

protected:

    char deptName[30];

    char assignedWork[30];

    int time2complete;

public:

    void getDeptInfo(void)

    {

        getBasicInfo(); //to get basic info of an employee

        cout << "Enter Department Name: ";

        cin.ignore(1);

        cin.getline(deptName, 30);

        cout << "Enter assigned work: ";

        fflush(stdin);

        cin.getline(assignedWork, 30);

        cout << "Enter time in hours to complete work: ";

        cin >> time2complete;

    }

    void printDeptInfo(void)

    {

        cout << "Employee's Information is: " << endl;

        cout << "Basic Information...:" << endl;

        cout << "Name: " << name << endl; //accessing protected data

        cout << "Employee ID: " << empId << endl; //accessing protected data

        cout << "Gender: " << gender << endl

             << endl; //accessing protected data

        cout << "Department Information...:" << endl;

        cout << "Department Name: " << deptName << endl; //accessing protected data

        cout << "Assigned Work: " << assignedWork << endl; //accessing protected data

        cout << "Time to complete work: " << time2complete << endl; //accessing protected data

    }

};

//another Base Class : loadInfo

class loanInfo : private basicInfo {

protected:

    char loanDetails[30];

    int loanAmount;

public:

    void getLoanInfo(void)

    {

        getBasicInfo(); //to get basic info of an employee

        cout << "Enter Loan Details: ";

        cin.ignore(1);

        cin.getline(loanDetails, 30);

        cout << "Enter loan amount: ";

        cin >> loanAmount;

    }

    void printLoanInfo(void)

    {

        cout << "Employee's Information is: " << endl;

        cout << "Basic Information...:" << endl;

        cout << "Name: " << name << endl; //accessing protected data

        cout << "Employee ID: " << empId << endl; //accessing protected data

        cout << "Gender: " << gender << endl

             << endl; //accessing protected data

        cout << "Loan Information...:" << endl;

        cout << "Loan Details: " << loanDetails << endl; //accessing protected data

        cout << "Loan Amount : " << loanAmount << endl; //accessing protected data

    }

};

int main()

{

    //read and print department information

    deptInfo objD;

    objD.getDeptInfo();

    objD.printDeptInfo();

    cout << endl

         << endl;

    //read and print loan information

    loanInfo objL;

    objL.getLoanInfo();

    objL.printLoanInfo();

    return 0;

}

2. Write a C++ Program to illustrates the

use of Constructors in

multilevel inheritance

code-

#include<iostream>

**using** **namespace** std;

// Base class

**class** A

{

**public**:

        A()

        {

            cout << "Base class A constructor \n";

        }

};

// Derived class B

**class** B: **public** A

{

**public**:

        B()

        {

            cout << "Class B constructor \n";

        }

};

// Derived class C

**class** C: **public** B

{

**public**:

        C()

        {

            cout << "Class C constructor \n";

        }

};

// Driver code

**int** main()

{

    C obj;

**return** 0;

}

3. Write a C++ Program to Overriding

member functions using

Inheritance.

Code-

#include <iostream>

using namespace std;

class Base {

public:

void print() {

cout << "Base Function" << endl;

}

};

class Derived : public Base {

public:

void print() {

cout << "Derived Function" << endl;

}

};

int main() {

Derived derived1;

derived1.print();

return 0;

}

4. Write a C++ Program to calculate the

total mark of a student

using the concept of inheritance and

virtual base class.

Code-

**#include<iostream.h>**

**#include<conio.h>**

class student {

int rno;

public:

void **getnumber**() {

cout << "Enter Roll No:";

cin>>rno;

}

void **putnumber**() {

cout << "\n\n\tRoll No:" << rno << "\n";

}

};

class test : virtual public student {

public:

int part1, part2;

void **getmarks**() {

cout << "Enter Marks\n";

cout << "Part1:";

cin>>part1;

cout << "Part2:";

cin>>part2;

}

void **putmarks**() {

cout << "\tMarks Obtained\n";

cout << "\n\tPart1:" << part1;

cout << "\n\tPart2:" << part2;

}

};

class sports : public virtual student {

public:

int score;

void **getscore**() {

cout << "Enter Sports Score:";

cin>>score;

}

void **putscore**() {

cout << "\n\tSports Score is:" << score;

}

};

class result : public test, public sports {

int total;

public:

void **display**() {

total = part1 + part2 + score;

putnumber();

putmarks();

putscore();

cout << "\n\tTotal Score:" << total;

}

};

void **main**() {

result obj;

clrscr();

obj.getnumber();

obj.getmarks();

obj.getscore();

obj.display();

getch();

}