**29 NOV , LAB-2**

1. **Create an empty class without any data member and member functions.**

**#include<iostream>**

**using namespace std;**

**class Empty{};**

**int main()**

**{**

**cout<<"The Class in this Program is Empty.";**

**}**

1. **WAP with class name as Student consisting of Roll, Name and Marks as private data members. Assign values in these members without using constructor but simple function. Also have a function to print these values.**

**#include<iostream>**

**using namespace std;**

**class Student{**

**string name;**

**int roll;**

**double marks;**

**public :**

**void getData()**

**{**

**cout<<"Enter your Name : ";**

**getline(cin,name);**

**cout<<"Enter your Roll Number : ";**

**cin>>roll;**

**cout<<"Enter your Marks : ";**

**cin>>marks;**

**}**

**void displayData()**

**{**

**cout<<"\n\tName : "<<name<<endl;**

**cout<<"\tRoll No. : "<<roll<<endl;**

**cout<<"\tMarks : "<<marks<<endl;**

**}**

**};**

**int main()**

**{**

**Student student1;**

**student1.getData();**

**student1.displayData();**

**}**

1. **WAP with class name as Teacher. Declare 3 methods named as printText1(), printText2() and printValue(int a) inside the class but define them outside the class definition. To test it, call these methods from the main function.**

**#include<iostream>**

**using namespace std;**

**class Teacher{**

**public :**

**void printText1();**

**void printText2();**

**void printValue(int a);**

**};**

**void Teacher::printText1(){**

**cout<<"\n\tThis is Text No.1.";**

**}**

**void Teacher::printText2(){**

**cout<<"\n\tThis is Text No.2.";**

**}**

**void Teacher::printValue(int a){**

**cout<<"\n\tThe Entered Number is "<<a<<".";**

**}**

**int main()**

**{**

**Teacher teacher1;**

**int num;**

**cout<<"\nEnter a number : ";**

**cin>>num;**

**teacher1.printText1();**

**teacher1.printText2();**

**teacher1.printValue(num);**

**}**

**4. WAP to create multiple objects of a class. Create a class named Laptop with appropriate member variables in public access specifier. Assign values and print these values from main function. [No need of any methods.]**

**#include<iostream>**

**using namespace std;**

**class Laptop{**

**public:**

**string brand\_name;**

**string model;**

**int price;**

**};**

**int main(){**

**Laptop laptop1;**

**Laptop laptop2;**

**laptop1.brand\_name="Acer";**

**laptop1.model="Nitro 5";**

**laptop1.price=110000;**

**laptop2.brand\_name="Dell";**

**laptop2.model="Inspiron 15 3593";**

**laptop2.price=89990;**

**cout<<"Brand Name : "<<laptop1.brand\_name<<endl;**

**cout<<"Model No. : "<<laptop1.model<<endl;**

**cout<<"Price : "<<laptop1.price<<endl;**

**cout<<"\nBrand Name : "<<laptop2.brand\_name<<endl;**

**cout<<"Model No. : "<<laptop2.model<<endl;**

**cout<<"Price : "<<laptop2.price<<endl;**

**}**

**5. Modify Program no. 4 keeping member variables as private. Have some public methods to input and output the values.**

**#include<iostream>**

**using namespace std;**

**class Laptop{**

**string brand\_name;**

**string model;**

**int price;**

**public :**

**void getData()**

**{**

**cout<<"\nEnter the Name of the Brand : ";**

**getline(cin,brand\_name);**

**cout<<"\nEnter the Model Name : ";**

**getline(cin,model);**

**cout<<"\nEnter the Price : ";**

**cin>>price;**

**}**

**void displayData()**

**{**

**cout<<"\n\tBrand Name : "<<brand\_name;**

**cout<<"\n\tModel Name : "<<model;**

**cout<<"\n\tPrice : "<<price;**

**}**

**};**

**int main()**

**{**

**Laptop laptop1;**

**laptop1.getData();**

**laptop1.displayData();**

**}**

**6. WAP to add odd numbers between 1 and 100 and display its sum using class.**

**#include<iostream>**

**using namespace std;**

**class Sum{**

**int lower\_range;**

**int upper\_range;**

**public:**

**void getData(int a,int b)**

**{**

**if(a>b)**

**{**

**lower\_range=b;**

**upper\_range=a;**

**}**

**else**

**{**

**lower\_range=a;**

**upper\_range=b;**

**}**

**}**

**int natural\_sum()**

**{**

**return upper\_range\*((upper\_range+1)/2.0);**

**}**

**int even\_sum()**

**{**

**return (upper\_range/2.0)\*((upper\_range/2.0)+1);**

**}**

**void odd\_sum()**

**{**

**cout<<"\tSum of all Odd Numbers between "<<lower\_range<<" and "<<upper\_range<<**

**" is "<<natural\_sum()-even\_sum()<<endl;**

**}**

**};**

**int main()**

**{**

**Sum sum1;**

**sum1.getData(1,100);**

**sum1.odd\_sum();**

**}**

**7. WAP using constructor and destructor to display student details. You are free to use any other concept as preferred to input and output information.**

**#include<iostream>**

**using namespace std;**

**class Student{**

**string name;**

**int roll;**

**int Class;**

**string address;**

**public :**

**Student()**

**{**

**cout<<"\nEnter the Name of the Student : ";**

**getline(cin,name);**

**cout<<"\nEnter the Address of the Student : ";**

**getline(cin,address);**

**cout<<"\nEnter the Roll No.of the Student : ";**

**cin>>roll;**

**cout<<"\nEnter the Class of the Student : ";**

**cin>>Class;**

**}**

**void displayData()**

**{**

**cout<<"\n\tName : "<<name;**

**cout<<"\n\tClass : "<<Class;**

**cout<<"\n\tRoll No. : "<<roll;**

**cout<<"\n\tAddress : "<<address;**

**}**

**~Student()**

**{**

**cout<<"\n\nEnd of the Program.";**

**}**

**};**

**int main()**

**{**

**Student student1;**

**student1.displayData();**

**}**

**8. WAP to find the volume of sphere, cube and cylinder using function overloading. [Volume of cylinder = πr2h, sphere = 4/3 πr3, cube = a3 ]**

**#include<iostream>**

**using namespace std;**

**float Volume(float r,float PI)**

**{**

**return (4/3.0)\*PI\*(r\*r\*r);**

**}**

**float Volume(float r,float h,float PI)**

**{**

**return PI\*r\*r\*h;**

**}**

**float Volume(float l)**

**{**

**return l\*l\*l;**

**}**

**int main()**

**{**

**float radius,height,length,PI=22/7.0;**

**cout<<"\nEnter the Radius of the Sphere : ";**

**cin>>radius;**

**cout<<"\nEnter the Height of the Cylinder : ";**

**cin>>height;**

**cout<<"\nEnter the Length of the Cube : ";**

**cin>>length;**

**cout<<"\n\tVolume of the Sphere : "<<Volume(radius,PI);**

**cout<<"\n\tVolume of the Cylinder : "<<Volume(radius,height,PI);**

**cout<<"\n\tVolume of the Cube : "<<Volume(length);**

**}**