ASSIGNMENT 2:

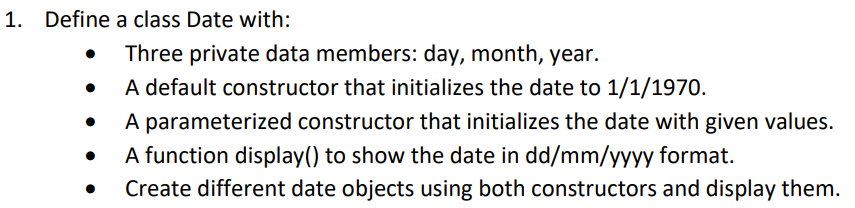
Name: Om Prasanna Kulkarni

PRN: 22510034

Batch: S3

Branch: CSE

Topic: Constructors, Destructors and Static Members



#include <iostream>

using namespace std;

class Date

{

private:

    int day, month, year;

public:

    Date()

    {

        cout << "Initial Date: 1/1/1970" << endl;

    }

    Date(int iday, int imonth, int iyear)

    {

        day = iday;

        month = imonth;

        year = iyear;

    }

    void display()

    {

        cout <<"Input Date: "<< day << "/" << month << "/" << year << endl;

    }

};

int main()

{

    int day, month, year;

    cout << "Enter the day: " << endl;

    cin >> day;

    cout << "Enter the month: " << endl;

    cin >> month;

    cout << "Enter the year: " << endl;

    cin >> year;

    Date object;

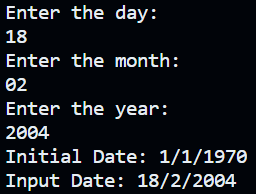
    Date object1(day, month, year);

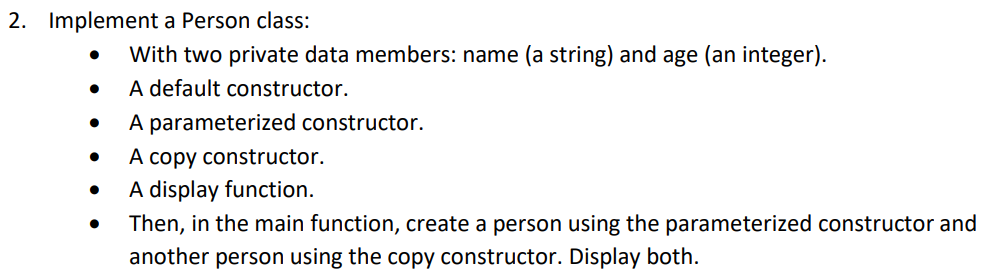
    object1.display();

    return 0;

}

OUTPUT:





#include<iostream>

#include<string>

using namespace std;

class Person{

    private:

    string name;

    int age;

    public:

    Person(){

        cout<<"This is the default constructor."<<endl;

    }

    Person(string iname,int iage){

      cout<<"Parametrized constructor called."<<endl;

      name=iname;

      age=iage;

    }

    Person(Person &t){

        cout<<"Copy constructor called."<<endl;

        name=t.name;

        age=t.age;

    }

void display(){

    cout<<"The name of the person is: "<<name<<" and the age is: "<<age<<endl;

}

};

int main(){

    string name;

    int age;

    cout<<"Enter the name of the person: "<<endl;

    cin>>name;

    cout<<"Enter the age of the person: "<<endl;

    cin>>age;

    Person object;

    Person object1(name,age);

    object1.display();

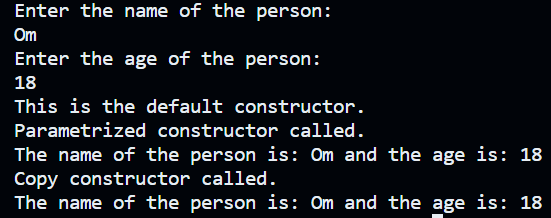
    Person object2=object1;

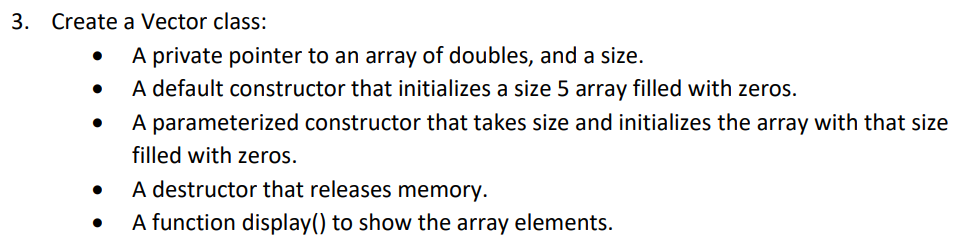
    object2.display();

    return 0;

}

OUTPUT:



#include<iostream>

#include<vector>

using namespace std;

class Vector{

    private:

    int size;

    double \*p;

public:

Vector(){

  size=5;

  p=new double[size];

  for(int i=0;i<size;i++){

    p[i]=0.00;

  }

}

Vector(int isize){

    size=isize;

    p=new double[size];

    for(int i=0;i<size;i++){

        p[i]=0.00;

    }

  }

~Vector(){

    delete[] p;

}

void display(){

    for(int i=0;i<size;i++){

        cout<<p[i]<<" ";

    }

}

};

int main(){

  int size;

  cout<<"Enter the size of the array: "<<endl;

  cin>>size;

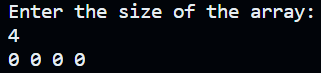
     Vector object(size);

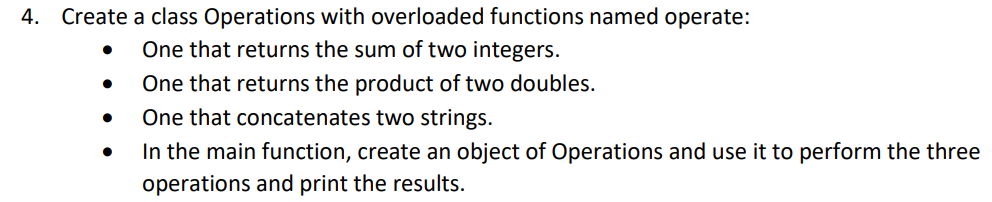
     object.display();

    return 0;

}

OUTPUT:





#include<iostream>

using namespace std;

class Operations{

   public:

   int a,b;

   double u,v;

   string p,q;

   void operate(int a,int b);

   void operate(double a,double b);

   void operate(string a,string b);

};

void Operations::operate(int ia,int ib){

   a=ia;

   b=ib;

   cout<<"The sum of integers is: "<<a+b<<endl;

}

void Operations::operate(double iu,double iv){

   u=iu;

   v=iv;

   cout<<"The product of double type numbers is: "<<u\*v<<endl;

}

void Operations::operate(string ip,string iq){

   p=ip;

   q=iq;

   cout<<"The concatenated string is: "<<p+q<<endl;

}

int main(){

    int a,b;

    double u,v;

    string p,q;

    Operations object;

    cout<<"Enter two integers to be added: "<<endl;

    cin>>a>>b;

    object.operate(a,b);

    cout<<endl;

    cout<<"Enter two double type numbers to be operated on: "<<endl;

    cin>>u>>v;

    object.operate(u,v);

    cout<<endl;

    cout<<"Enter two strings to be concatenated (separated by a space): "<<endl;

    cin>>p>>q;

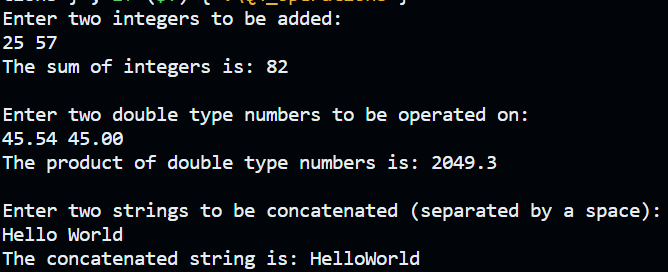
    object.operate(p,q);

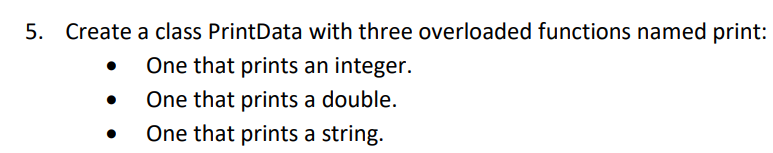
    cout<<endl;

    return 0;

}

OUTPUT:





#include<iostream>

using namespace std;

class printData{

   public:

   int x;

   double y;

   string z;

   void print(int x);

   void print(double y);

   void print(string z);

};

void printData::print(int ix){

     x=ix;

     cout<<"The integer entered was: "<<x<<endl;

}

void printData::print(double iy){

     y=iy;

     cout<<"The double typed number entered was: "<<y<<endl;

}

void printData::print(string iz){

     z=iz;

     cout<<"The string entered was: "<<z<<endl;

}

int main(){

    int x;

    double y;

    string z;

    printData obj;

    cout<<"Enter an integer: "<<endl;

    cin>>x;

    obj.print(x);

    cout<<"Enter a double type number: "<<endl;

    cin>>y;

    obj.print(y);

    cout<<"Enter a string: "<<endl;

    cin.ignore();

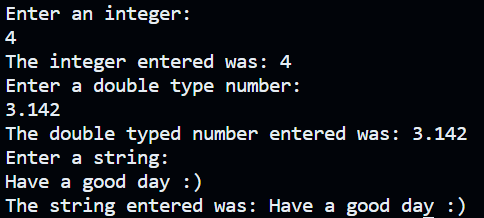
    getline(cin,z);

    obj.print(z);

    return 0;

}

OUTPUT:



6. Create a Team class that manages team members up to a fixed maximum size (e.g., 10). The class should have:

#include<iostream>

using namespace std;

class Team{

    string arr[10];

    int counter=0;

 public:

 string name;

 Team(){

      for(int i=0;i<10;i++){

        arr[i]="0";

      }

   }

 void addMember(string name);

 void addMember();

 void displayTeam();

};

void Team::addMember(string iname){

       if(counter>=10){

        cout<<"The team has been completed."<<endl;

    }

    else{

        arr[counter]=iname;

        counter++;

 }

}

void Team::addMember(){

    if(counter>=10){

        cout<<"The team has been completed already."<<endl;

    }

    else{

        cout<<"\n\n             Team with Generic names of Members: "<<endl;

        for(int i=0;i<10;i++){

            arr[i]="Unknown"+to\_string(i+1);

            counter++;

        }

    }

}

void Team::displayTeam(){

  for(int i=0;i<10;i++){

    cout<<arr[i]<<" ";

  }

}

int main(){

    string name;

    Team obj;

    Team obj1;

    cout<<"Enter the name of team members for a team of 10 persons (Separated by spaces): "<<endl;

    for(int i=0;i<10;i++){

    cin>>name;

    obj.addMember(name);

    }

    cout<<"\n\n             Your Entered Team is: \n";

    obj.displayTeam();

    obj1.addMember();

    obj1.displayTeam();

    return 0;

}

OUTPUT:

