**Q1:**

1. Base class of class **Heavyvehicle :** Vehicle

Derived class of class **Heavyvehicle : Bus**

1. data member(s) that can be accessed from function **displaydata() :**

No data member can be accessed because all its own and inherited data members are **private**.

1. Since, All its own and inherited data members are private, so no data member can be directly accessed until you use functions to access them.
2. Since, **outputdata()** is made protected when inherited so it is not directly accessible to the objects of **Heavyvehicle** class but can be accessed using functions.

**Q2:**

**Code:**

#include<iostream>

#include<string>

using namespace std;

class wheel

{

private:

string w;

public:

void set\_wheel\_state(string s)

{

w=s;

}

string get\_wheel\_state()

{

return w;

}

};

class car: public wheel

{

private:

string w[4];

public:

void set\_car\_to\_moving()

{

for(int i=0;i<4;i++)

{

w[i]="Turning";

}

}

void set\_car\_to\_stopped()

{

for(int i=0;i<4;i++)

{

w[i]="Stopped";

}

}

void print\_car\_wheels\_state()

{

cout<<"Car state:- "<<endl;

for(int i=0;i<4;i++)

{

cout<<"Wheel "<<i<<" is "<<w[i]<<endl;

}

}

};

int main()

{

car a;

string x;

cout<<"Enter wheel state : (moving/stopped)"<<endl;

cin>>x;

a.set\_wheel\_state(x);

cout<<"Wheel state : "<<a.get\_wheel\_state()<<endl;

if(a.get\_wheel\_state()=="moving")

{

a.set\_car\_to\_moving();

a.print\_car\_wheels\_state();

}

else if(a.get\_wheel\_state()=="stopped")

{

a.set\_car\_to\_stopped();

a.print\_car\_wheels\_state();

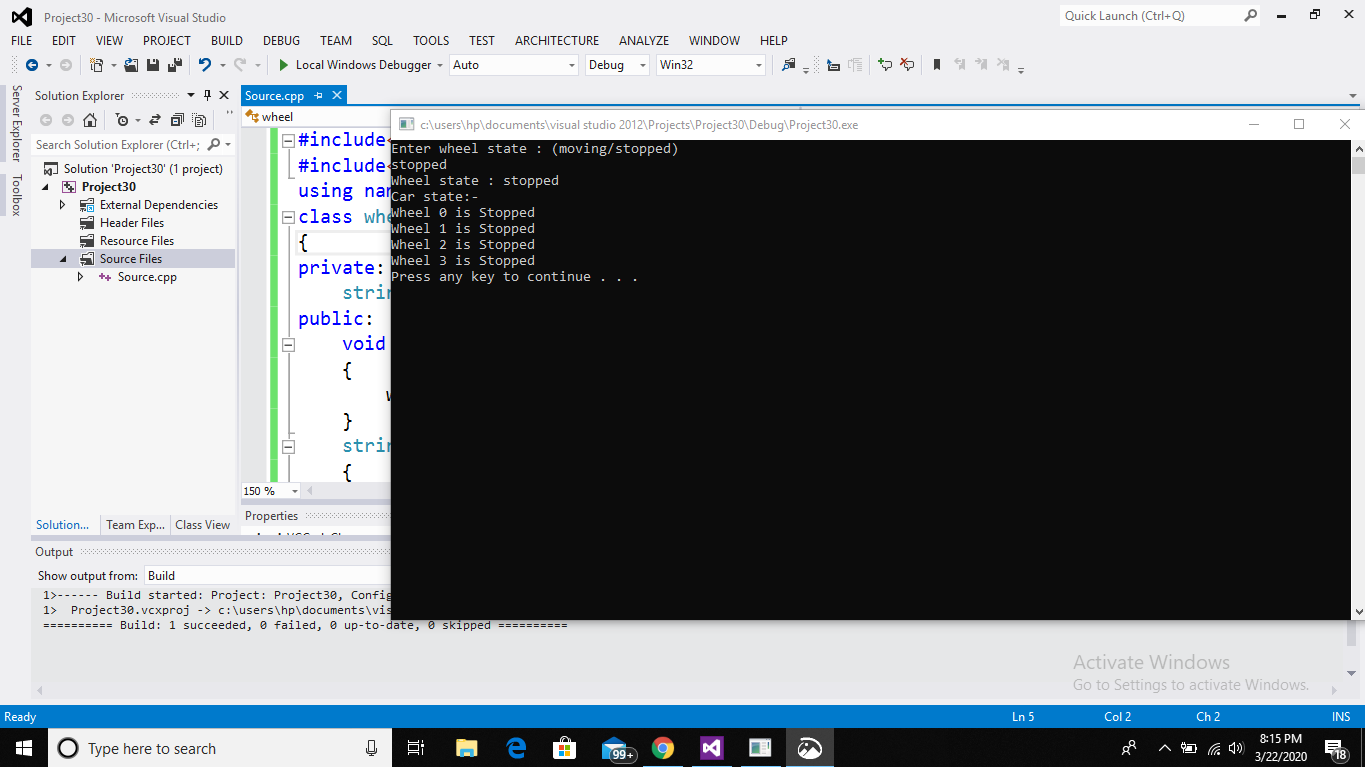
}

system("pause");

return 0;

}

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Q3:**

**Code:**

#include<iostream>

#include<string>

using namespace std;

class person

{

public:

string name;

int age;

void set\_values() //set function to take input and save record of name and age

{

string x;

cout<<"Enter name :";

cin>>x;

name=x;

int c;

cout<<"Enter age :";

cin>>c;

age=c;

}

void change\_values() //function to change values of name or age according user's consent

{

string a;

char s='n';

cout<<"Do u want to change name or age? (y/n) ";

cin>>s;

if(s=='y')

{

cout<<"What do u want to change name or age? ";

cin>>a;

if(a=="name")

{

string x;

cout<<"Enter new name :";

cin>>x;

name=x;

cout<<"Name has been changed successfuly. "<<endl;

string v;

cout<<"do u also want to change age?(yes/no) ";

cin>>v;

if(v=="yes")

{

int x;

cout<<"Enter new age :";

cin>>x;

age=x;

cout<<"Age has been changed successfuly. "<<endl;

}

else if(v=="no")

{

}

}

else if(a=="age")

{

int x;

cout<<"Enter new age :";

cin>>x;

age=x;

cout<<"Age has been changed successfuly. "<<endl;

string v;

cout<<"do u also want to change name?(yes/no) ";

cin>>v;

if(v=="yes")

{

string x;

cout<<"Enter new name :";

cin>>x;

name=x;

cout<<"Name has been changed successfuly. "<<endl;

}

else if(v=="no")

{

}

}

}

else

{

}

}

void print\_address(person p) //function to print the address of respective object

{

cout<<"address : "<< this<<endl;

}

void del\_record()

{

char x='n';

cout<<" Do u want to delete (his/her) record? (y/n) ";

cin>>x;

if(x=='y')

{

name=" ";

age=0;

cout<<" --Record is deleted-- "<<endl;

}

else

{}

}

};

class student: public person //student class inheriting properties of person class

{

};

class employee: public person //employee class inheriting properties of person class

{

};

int main()

{

cout<<"For base person class : "<<endl;

person p[2];

for(int i=0;i<2;i++)

{

cout<<"for "<<i+1<<" person : "<<endl;

p[i].set\_values();

p[i].change\_values();

p[i].del\_record();

p[i].print\_address(p[i]);

cout<<endl;

}

cout<<endl<<"For derived student class : "<<endl;

student s[2];

for(int i=0;i<2;i++)

{

cout<<"for "<<i+1<<" student : "<<endl;

s[i].set\_values();

s[i].change\_values();

s[i].del\_record();

s[i].print\_address(s[i]);

cout<<endl;

}

cout<<endl<<"For derived employee class : "<<endl;

employee e[2];

for(int i=0;i<2;i++)

{

cout<<"for "<<i+1<<" employee : "<<endl;

e[i].set\_values();

e[i].change\_values();

e[i].del\_record();

e[i].print\_address(e[i]);

cout<<endl;

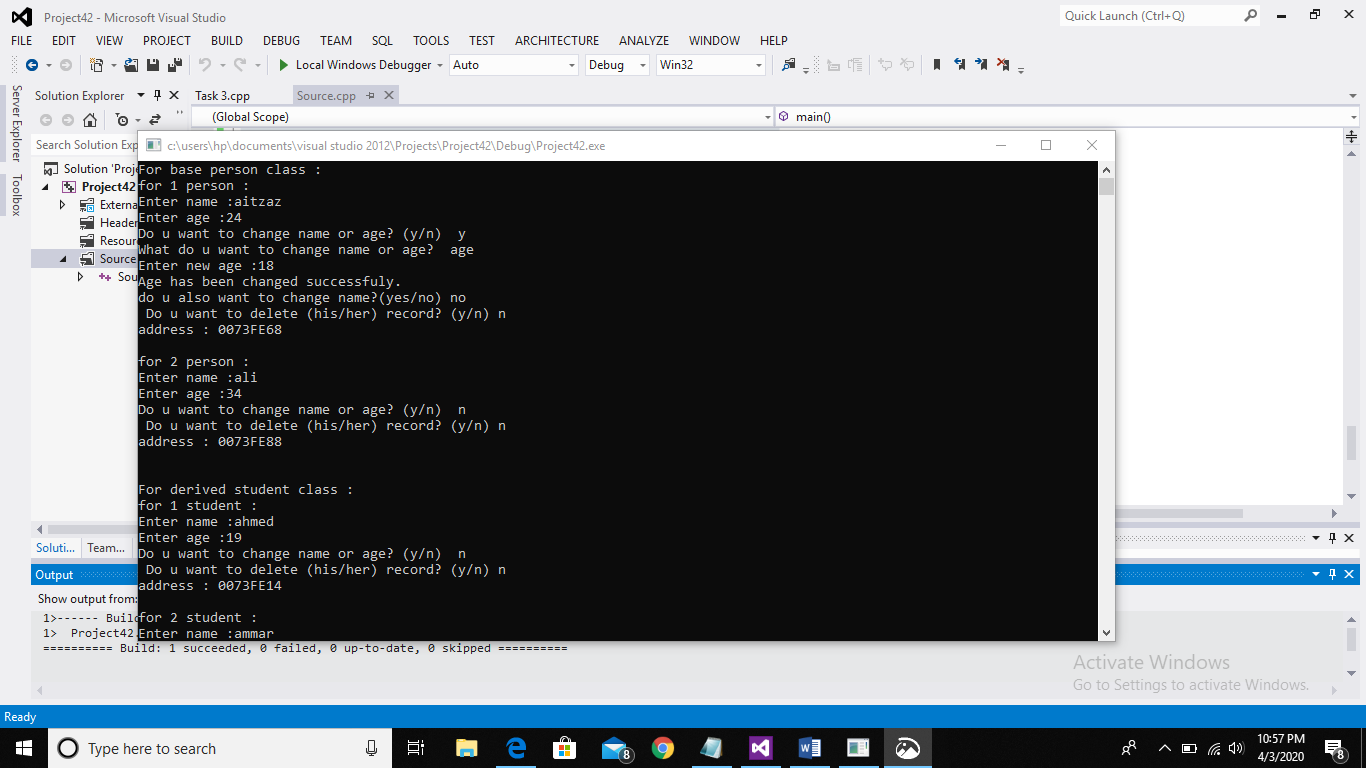
}

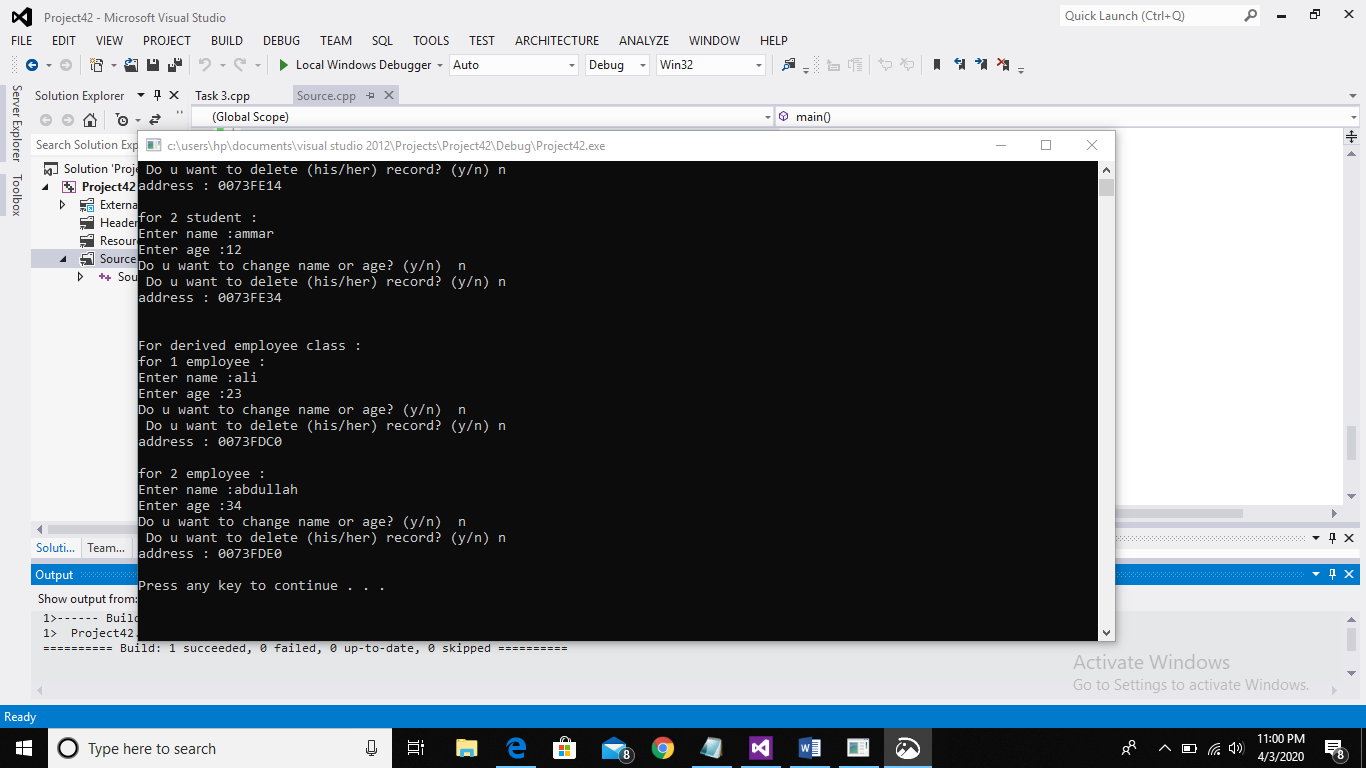
system("pause");

return 0;

}

**Output:**

****

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Q4:**

**Code:**

#include<iostream>

#include<string>

using namespace std;

class base

{

friend class b;

public:

int real,imag;

void first\_num() //function to enter first complex number

{

cout<<"Enter real and imaginary part of a complex number of base class : "<<endl;

cout<<"real : ";

cin>>real;

cout<<"imag : ";

cin>>imag;

cout<<endl;

}

};

class derived:public base

{

friend class b; //making class b friend of derived class

public:

int real1,imag1;

int areal,aimag;

void second\_num() //function to enter second complex number

{

cout<<"Enter real and imaginary part of a complex number of derived class : "<<endl;

cout<<"real : ";

cin>>real1;

cout<<"imag : ";

cin>>imag1;

cout<<endl;

}

void addition() //function for addition of both complex numbers

{

areal=real+real1;

aimag=imag+imag1;

if(aimag<0)

{

cout<<"complex number after addition : "<<areal<<aimag<<"i"<<endl;

}

else

{

cout<<"complex number after addition : "<<areal<<"+"<<aimag<<"i"<<endl;

}

}

void difference(derived d);

};

class b

{

public:

void difference(derived d) //function for difference of both complex numbers

{

d.areal=d.real-d.real1;

d.aimag=d.imag-d.imag1;

if(d.aimag<0)

{

cout<<"complex number after subtraction : "<<d.areal<<d.aimag<<"i"<<endl;

}

else

{

cout<<"complex number after subtraction : "<<d.areal<<"+"<<d.aimag<<"i"<<endl;

}

}

};

int main()

{

derived d;

d.first\_num();

d.second\_num();

cout<<"----Addition by base and inherited derived class----"<<endl;

d.addition();

cout<<endl<<"----Subtraction by friend class----"<<endl;

b b;

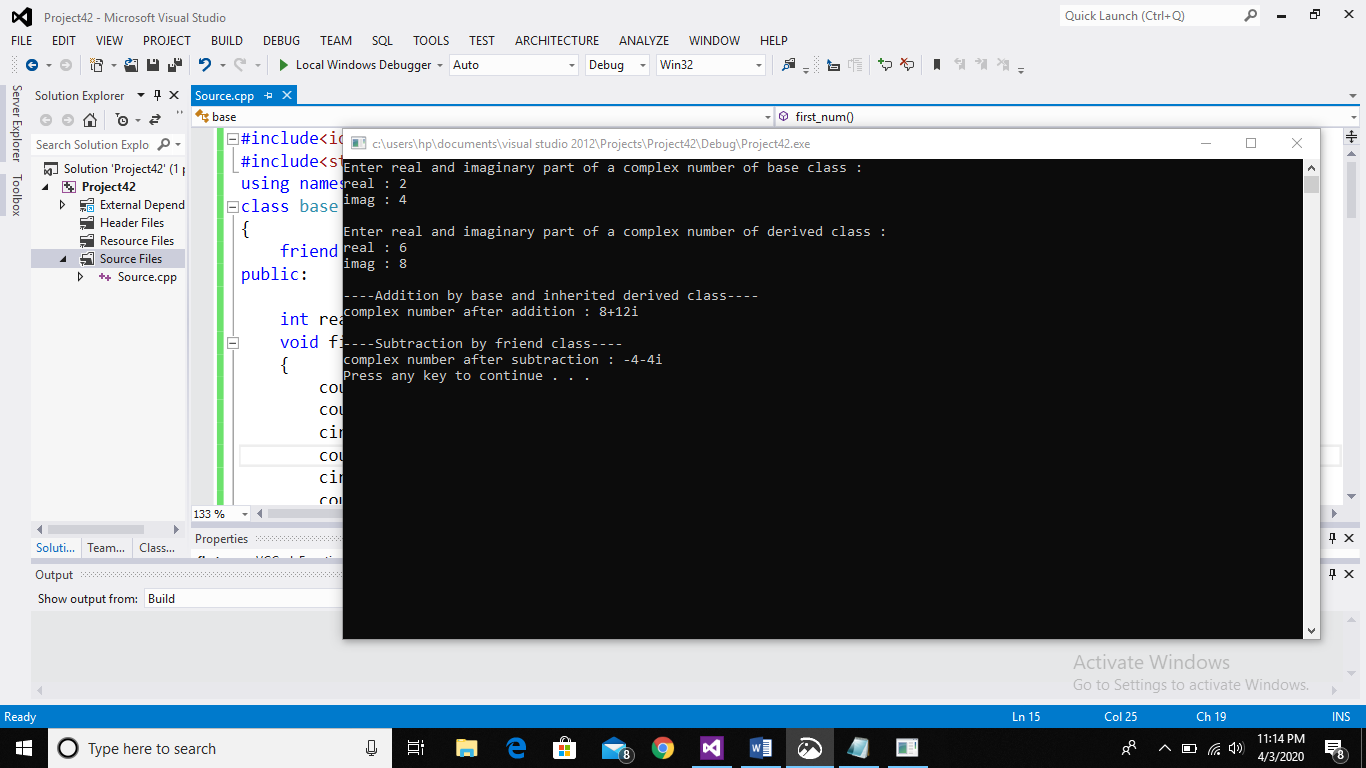
b.difference(d);

system("pause");

return 0;

}

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**