**9921103163**

**Nitin Chaudhary**

**F8**

**Ans-1-**

#include<iostream>

using namespace **std**;

class **book**{

    private:

    int book\_num;

**string** author;

**string** publisher;

    float price;

    int num\_copy\_issued;

    int num\_copy;

    public:

    void **fun**(int b,**string** a,**string** p,float pr,int issued,int num){

    book\_num=b;

    author**=**a;

    publisher**=**p;

    price=pr;

    num\_copy\_issued=issued;

    num\_copy=num;

    }

    void **issue**(**book** obj){

            if(obj.num\_copy>0){

                cout**<<**"The book written by "**<<**obj.author**<<**" is  available , you can issue this book\n";

                cout**<<**"Enter your name to issue this book";

**string** name;

                cin**>>**name;

                cout**<<**"\nThe book written by "**<<**obj.author**<<**" is issued on the accout of "**<<**name;

            }

            else{

                cout**<<**"\nhe book written by "**<<**obj.author**<<**" is  not available , you can issue this book , please choose another boook";

            }

    }

void **return\_book**(){

**string** bname;

**string** name;

    cout**<<**"\n\nReturning the book";

    cout**<<**"\n\nEnter book's author name to return";

**getchar**();

    cin**>>**bname;

    cout**<<**"\nEnter your name";

    cin**>>**name;

    cout**<<**"\nThe book written by "**<<**bname**<<**" is returned successfully by "**<<**name;

}

void **display**(**book** object){

    cout**<<**"\n\n\nDisplaying book information";

    cout**<<**"\nBook number is "**<<**object.book\_num;

    cout**<<**"\nBook author is "**<<**object.author;

    cout**<<**"\nBook publisher is "**<<**object.publisher;

    cout**<<**"\nBook price is "**<<**object.price;

    cout**<<**"\nNumber of copies issued are "**<<**object.num\_copy\_issued;

    cout**<<**"\nNumber of copies availabe are "**<<**object.num\_copy;

}

};

int **main**()

{

**book** obj1,obj2;

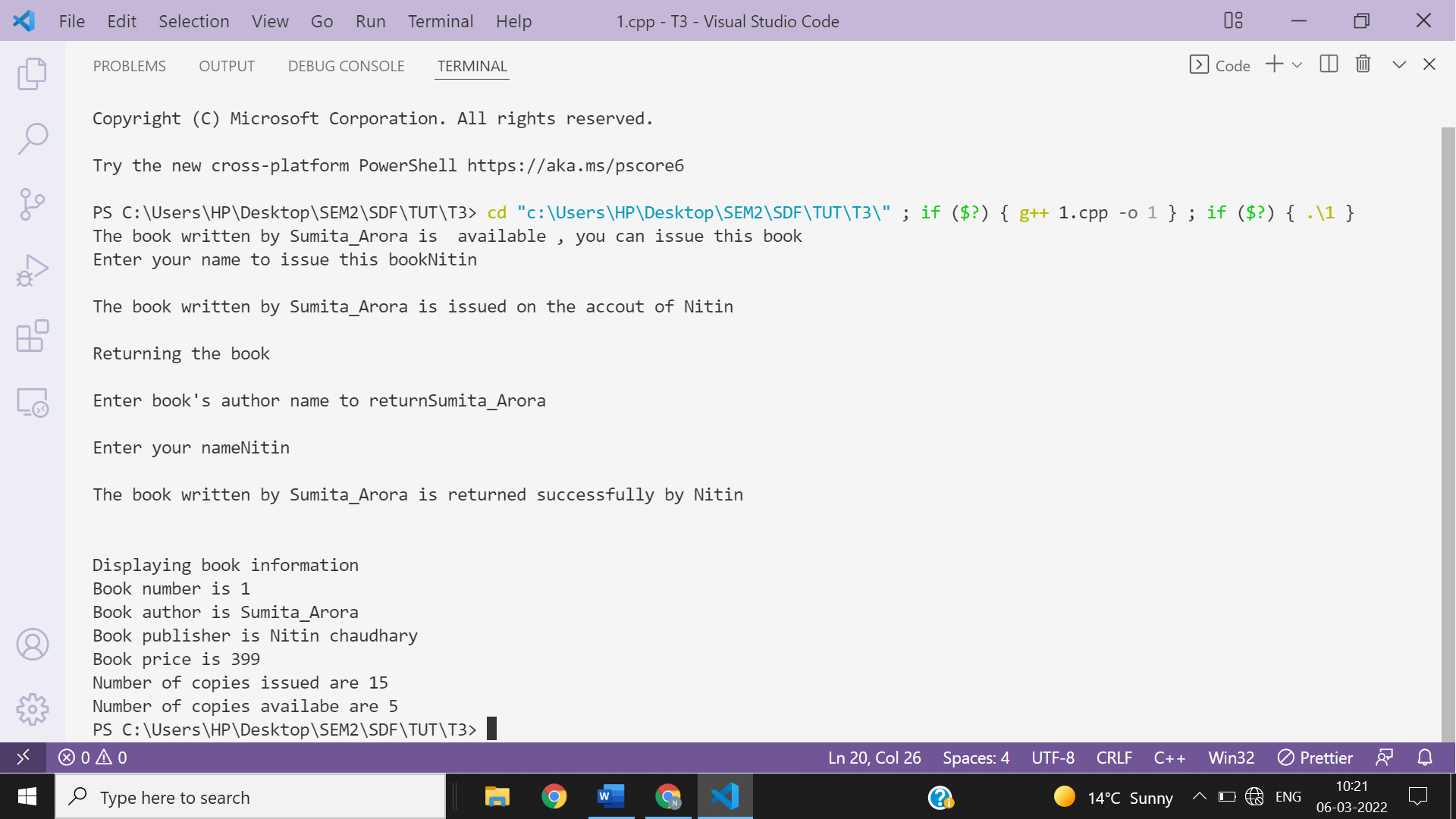
    obj1.**fun**(1,"Sumita\_Arora","Nitin chaudhary",399.0,15,5);

    obj2.**issue**(obj1);

    obj2.**return\_book**();

    obj2.**display**(obj1);

}



**Ans-2-**

#include <iostream>

#include<string>

using namespace **std**;

class **bank**

{

private:

**string** name;

    int num;

    char typ;

    float balance;

public:

    void **fun**(**string** nm, int n, char t, float b)

    {

        name **=** nm;

        num = n;

        typ = t;

        balance = b;

    }

    void **deposit**()

    {

        int n;

        cout **<<** "Enter the money you want deposit\n";

        cin **>>** n;

        balance += n;

        cout **<<** "Rs." **<<** n **<<** "deposited successfullly in the account of " **<<** name **<<** **endl**;

        cout **<<** "Available balance is " **<<** balance **<<** **endl**;

    }

    void **withdraw**()

    {

        if (balance > 1000)

        {

            cout **<<** "\nYou can withdraw max up to Rs. " **<<** balance - 1000 **<<** **endl**;

            int take;

            cout **<<** "Enter how much money you want to withdraw\n";

            cin **>>** take;

            if (take > balance-1000)

            {

                cout **<<** "Insufficient balance !!\n";

            }

            else

            {

                cout **<<** "Rs." **<<** take **<<** "withdrawn successfullly from the account of " **<<** name **<<** **endl**;

                balance=balance-take;

                cout **<<** "Available balance is " **<<** balance  **<<** **endl**;

            }

        }

        else

        {

            cout **<<** "You can not withdraw money because availabe balance is Rs." **<<** balance **<<** " only ,which are less than or equal to Rs. 1000";

        }

    }

    void **display**()

    {

        cout **<<** "\nDisplaying account details\n";

        cout **<<** "Name of account holder is: " **<<** name **<<** **endl**;

        cout **<<** "Account number is " **<<** num **<<** **endl**;

        cout **<<** "Type of account is " **<<** typ **<<** **endl**;

        cout **<<** "Available balance is " **<<** balance **<<** **endl**;

    }

};

int **main**()

{

**bank** account[2];

**string** nm;

    int n;

    char t;

    float b;

    for (int i = 0; i < 2; i++)

    {

        cout **<<** "\n\n\nEnter details for account " **<<** i+1;

        cout **<<** "\nName of account holder :\n";

**getchar**();

**getline**(cin,nm);*// function of string header file*

        cout **<<** "Account number :\n";

        cin **>>** n;

        cout **<<** "Type of account(S for saving and C for currents)\n";

        cin **>>** t;

        account[i].**fun**(nm, n, t, b);

        account[i].**deposit**();

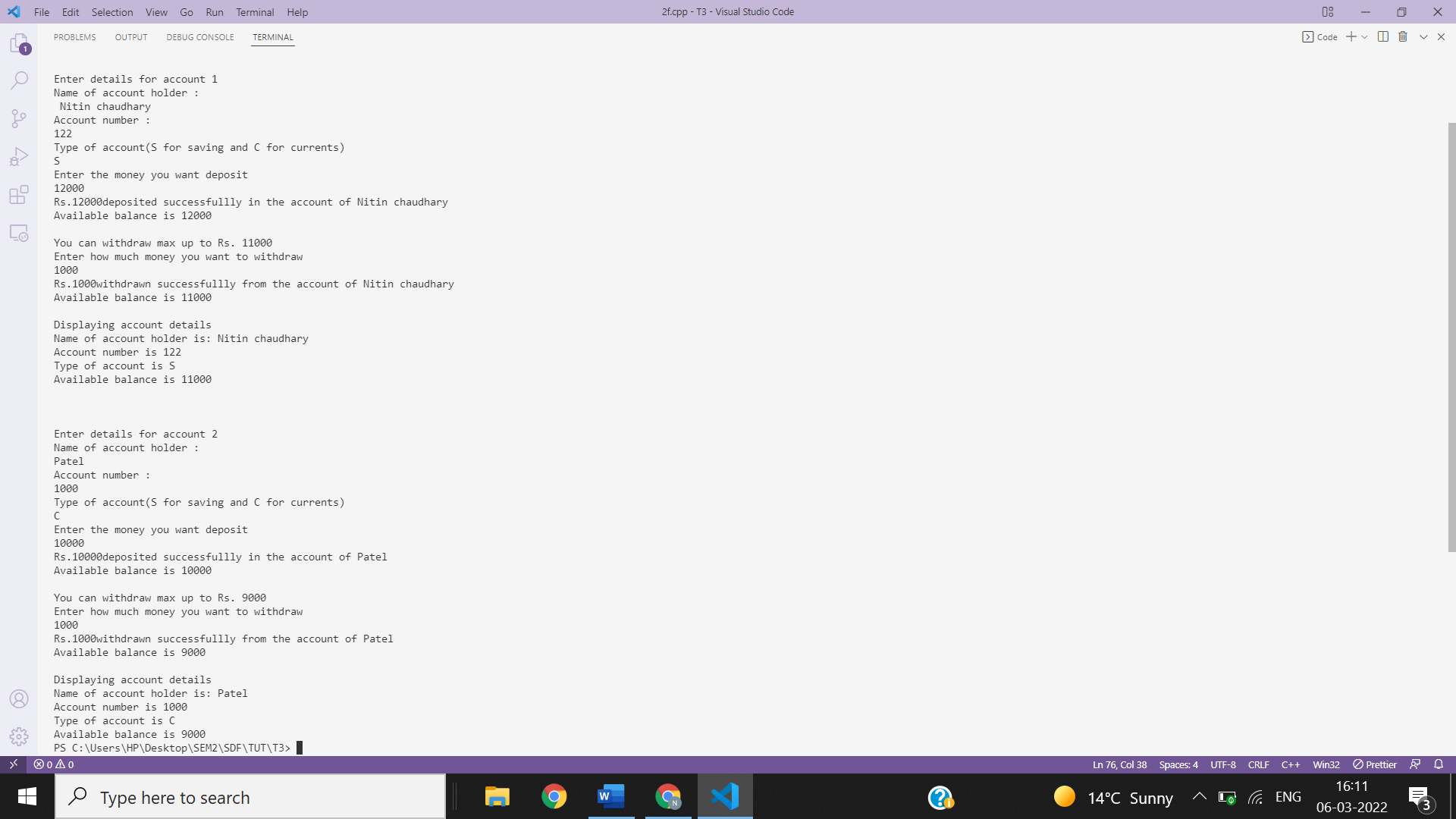
        account[i].**withdraw**();

        account[i].**display**();

    }

    return 0;

}



**Q3. What are static members of a class? When and how are they useful?**

**Ans 3-A member function that accesses only static data members of a class is static member functions. It cannot access other data members but only static members. The static data members are useful when some data values are to be shared across objects of the same class.**

**In case two or more than two objects , if static varible is changed by one object then this change will be reflected in all the objects i.e.; all objects will show same changed value because static variable is like a global variable and is available to all methods**.

**As we know that variables can be initialised only if it is defined completely i.e.; we must define static variable to initialise it or we must provide space to static variable to initialize it , it is done by the given syntax-**

**Datatype classname:: variableName;**

**But it is not the case with static variables defined and assigned simultaneously inside normal function , In this case we can access its same values which is assigned in the previous function call i.e;**

**Static int a=10; will remain 10 always when defined inside normal function.**

**Illustration code-**

**a-**

#include <iostream>

using namespace **std**;

class **Test**

{

public:

    static void **show**()

    {

        cout **<<** "This istest program for static functions!";

    }

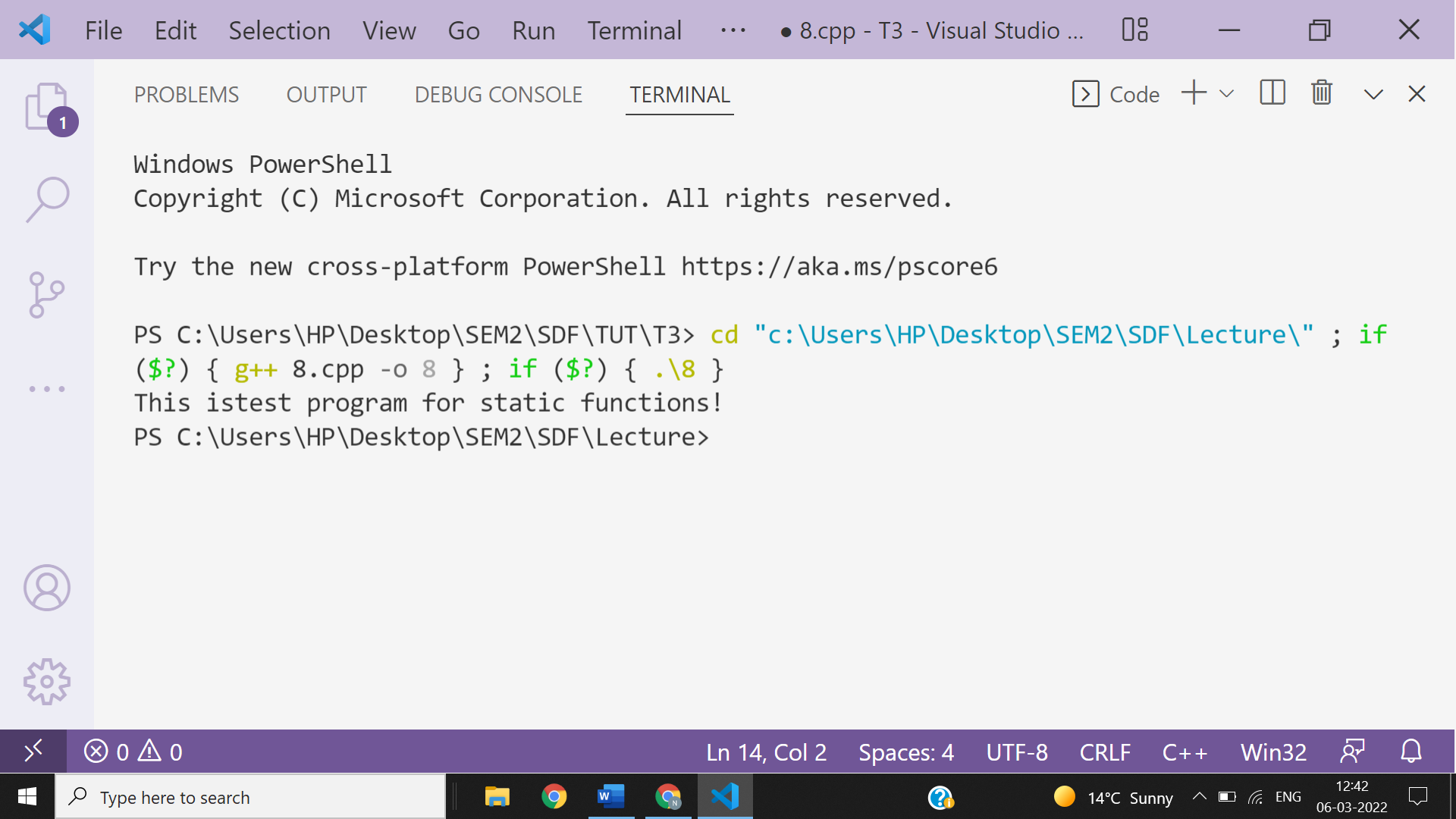
};

int **main**()

{

**Test**::**show**();

}



**b-**

#include <iostream>

using namespace **std**;

class **nitin**

{

private:

static int a;

public:

void **dis1**(**nitin** obj){

    cout**<<**obj.a;

}

static void **dis2**()

{

    cout**<<**a;

}

void **dis3**()

{

    cout**<<**a;

}

};

int **nitin**::a=3;

int **main**(){

**nitin** object,obj;

    obj.**dis1**(object);

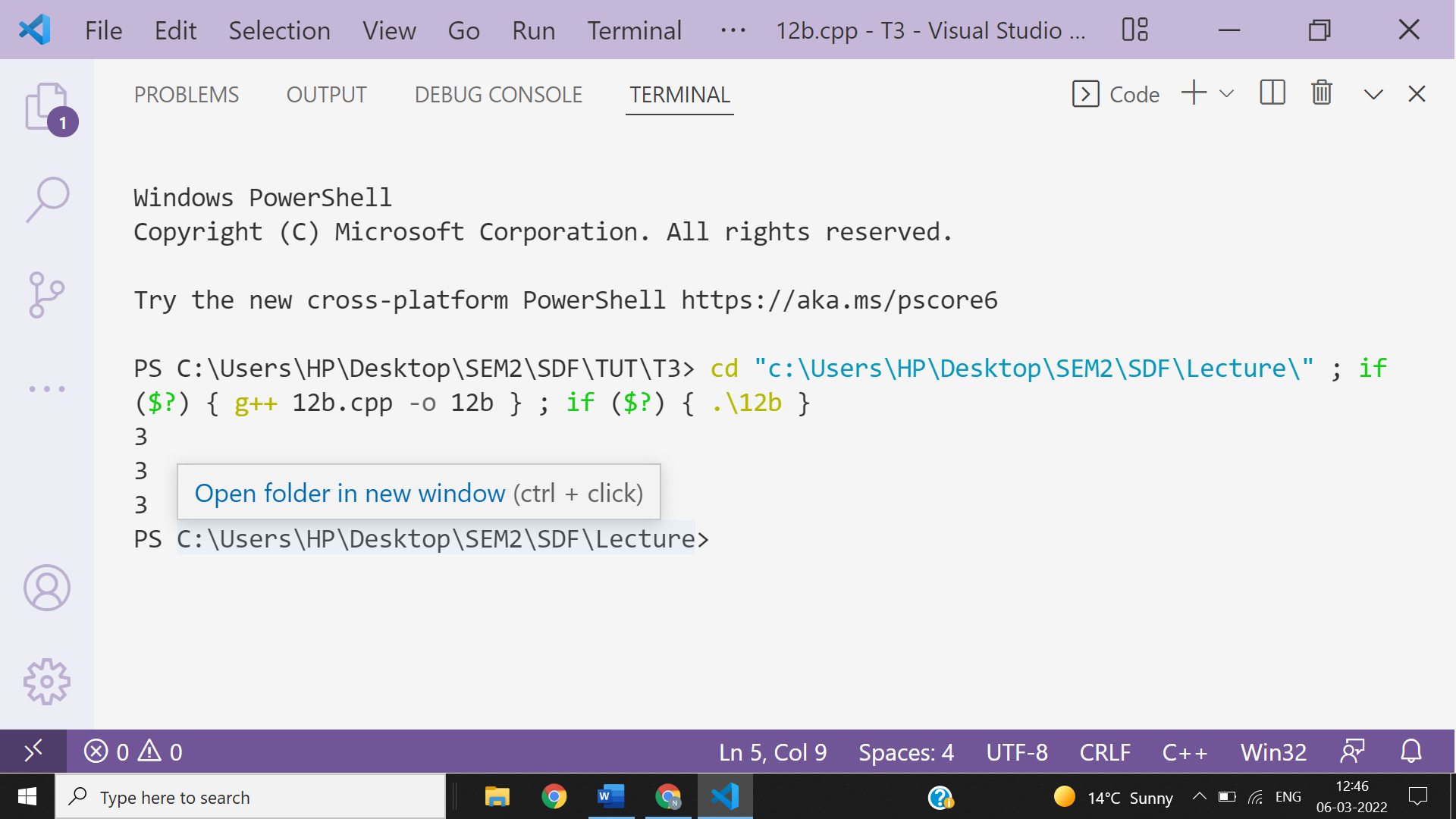
    cout**<<**"\n";

    obj.**dis2**();

    cout**<<**"\n";

    obj.**dis3**();

}



**c-**

#include<iostream>

using namespace **std**;

class **nitin**{

    private:

    static int a;

    int b;

    public:

**nitin**(int m){

        b=m;}

    void **fun**(int i){

        a=i;

        static int f=i;

        cout**<<**f;

    }

    void **print**()

    {

        cout**<<**a**<<**b;

    }

};

int **nitin**::a;

int **main**()

{

**nitin** **ob**(2);

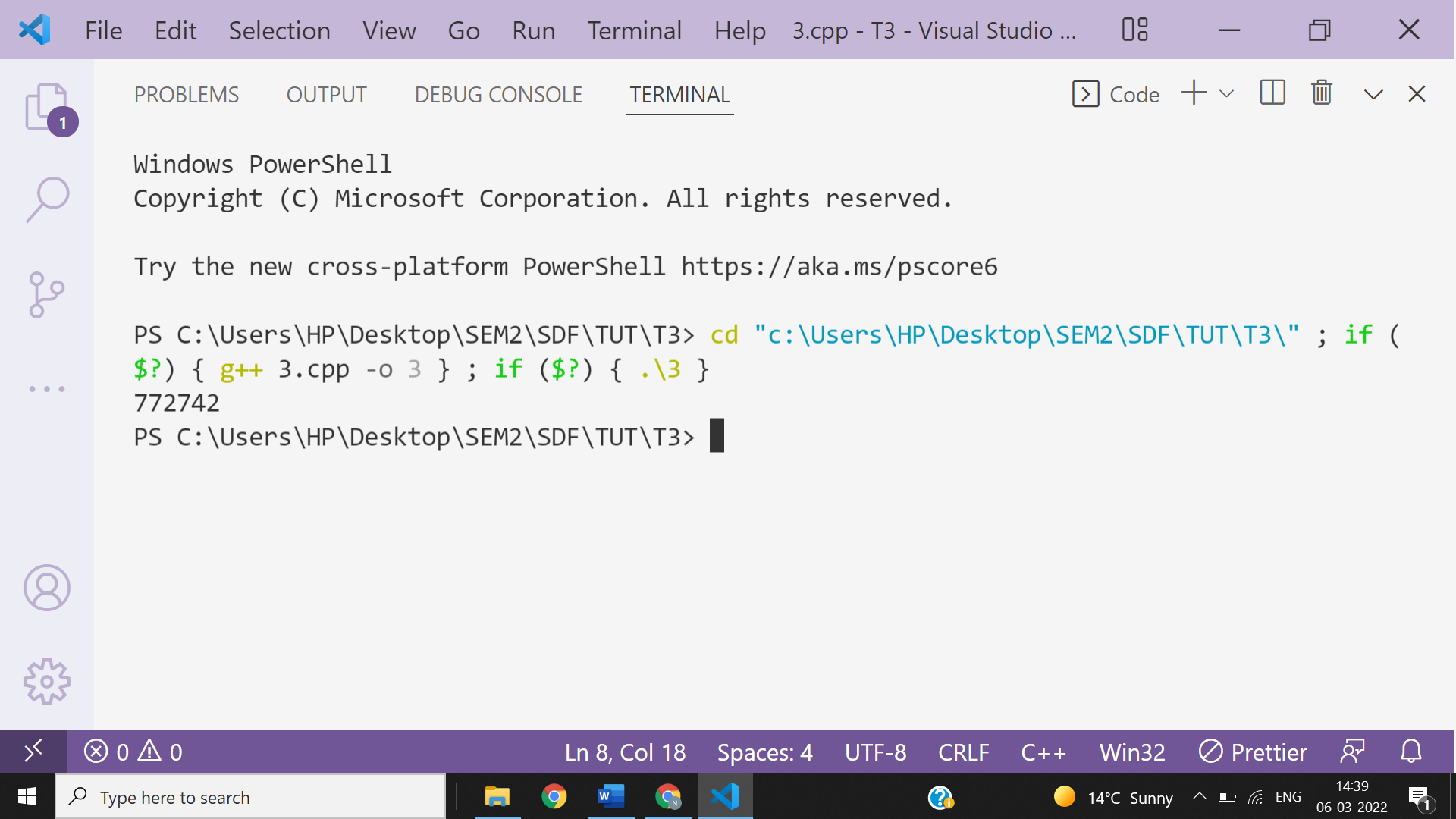
    ob.**fun**(7);

    ob.**print**();

    ob.**fun**(4);

    ob.**print**();

}



**d-**

#include<iostream>

using namespace **std**;

class **nitin**{

    private:

    static int a;

    int b;

    public:

**nitin**(int m){

        b=m;}

    void **fun**(int i){

        a=i;

        static int f;

        f=i;*//we can't write this in the above line due to re initialization of static variable f is not posssible.*

        cout**<<**f**<<endl**;

        cout**<<**&f**<<endl**;

*//f++;*

    }

    void **print**()

    {

        cout**<<**a**<<**b;

    }

};

int **nitin**::a;

int **main**()

{

**nitin** **ob**(2);

    ob.**fun**(7);

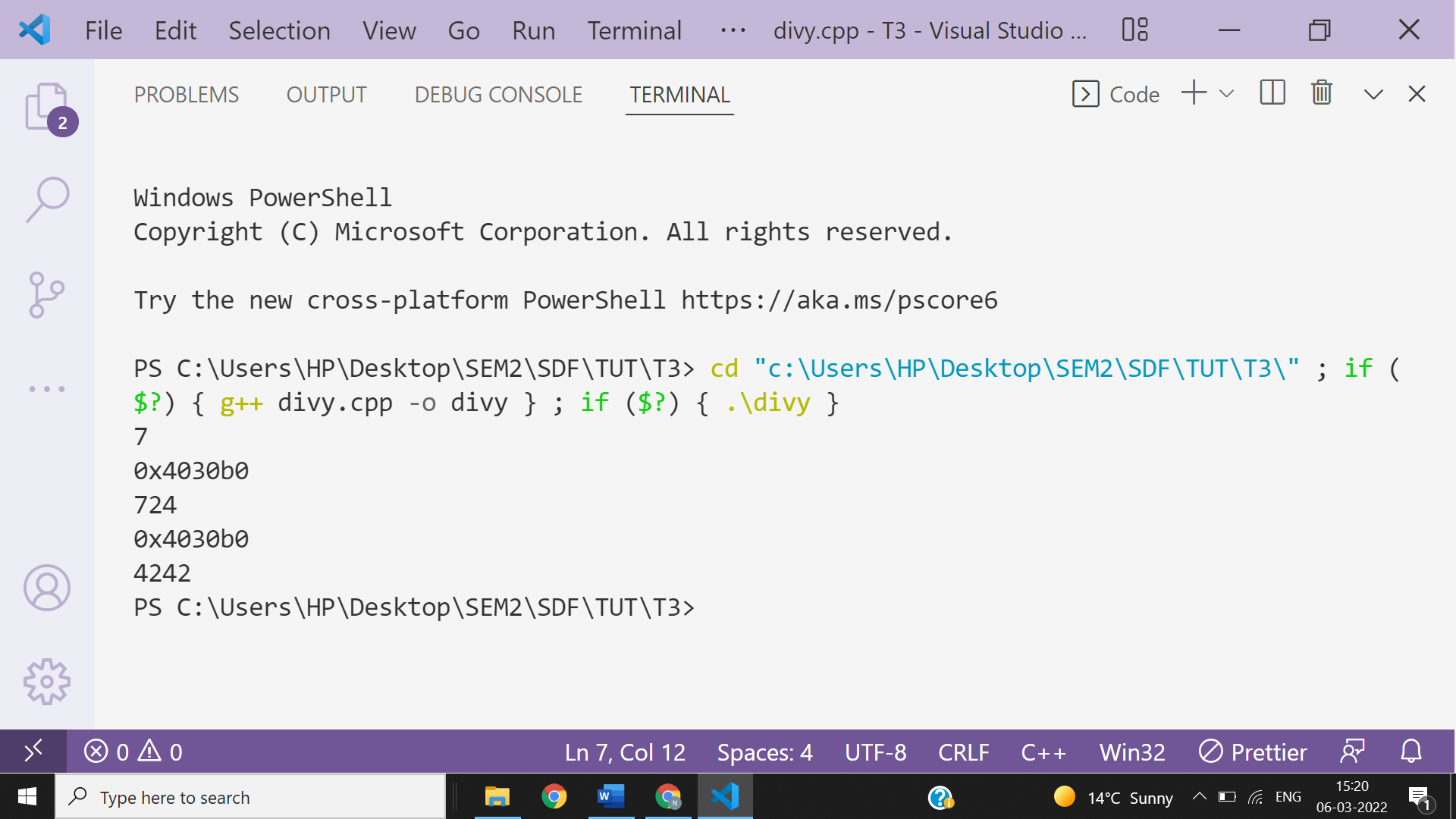
    ob.**print**();

    ob.**fun**(4);

    ob.**print**();

    ob.**print**();

}



**4-**

class **SmallObj**

{

private:

int some,more;

void **err\_1**() {cout<<"error";}

public:

void **Xdata**(int d) {some=d;more=d++; }

void **Ydata**() {cout<<some<<" "<<more; }

};

1. **Write the name that specifies the above class.**

**Ans- SmallObj**

1. **Write the data of the class with their access scope.**

**Ans-some and more**

**Scope is Private i.e.; only member and friend function of class can access these.**

**(iii) Write all member functions of the class along with their access**

**scope.**

**Ans-Xdata and Ydata**

**Scope is public i.e.; all class objects can access them.**

1. **Indicate the member function of the SmallObj that sets data.**

**Ans-Xdata is the member function which sets data.**