**9921103163**

**NITIN CHAUDHARY**

**F8**

**Week 5 Lab**

**Ans 1-**

#include<iostream>

using namespace **std**;

class **A**

{

public:

float Base;

void **getBase**(){

 cout**<<**"Enter value of base:";

 cin**>>**Base;

}

};

class **B**

{

public:

float Height;

void **getHeight**(){

 cout**<<**"Enter value of height:";

 cin**>>**Height;

}

};

class **C**:public **A**,public **B**{

float area;

public:

void **calArea**(){

    area=0.5\*Base\*Height;

    cout**<<**"Area of triangle is :"**<<**area;

}

};

int **main**()

{

**C** obj;

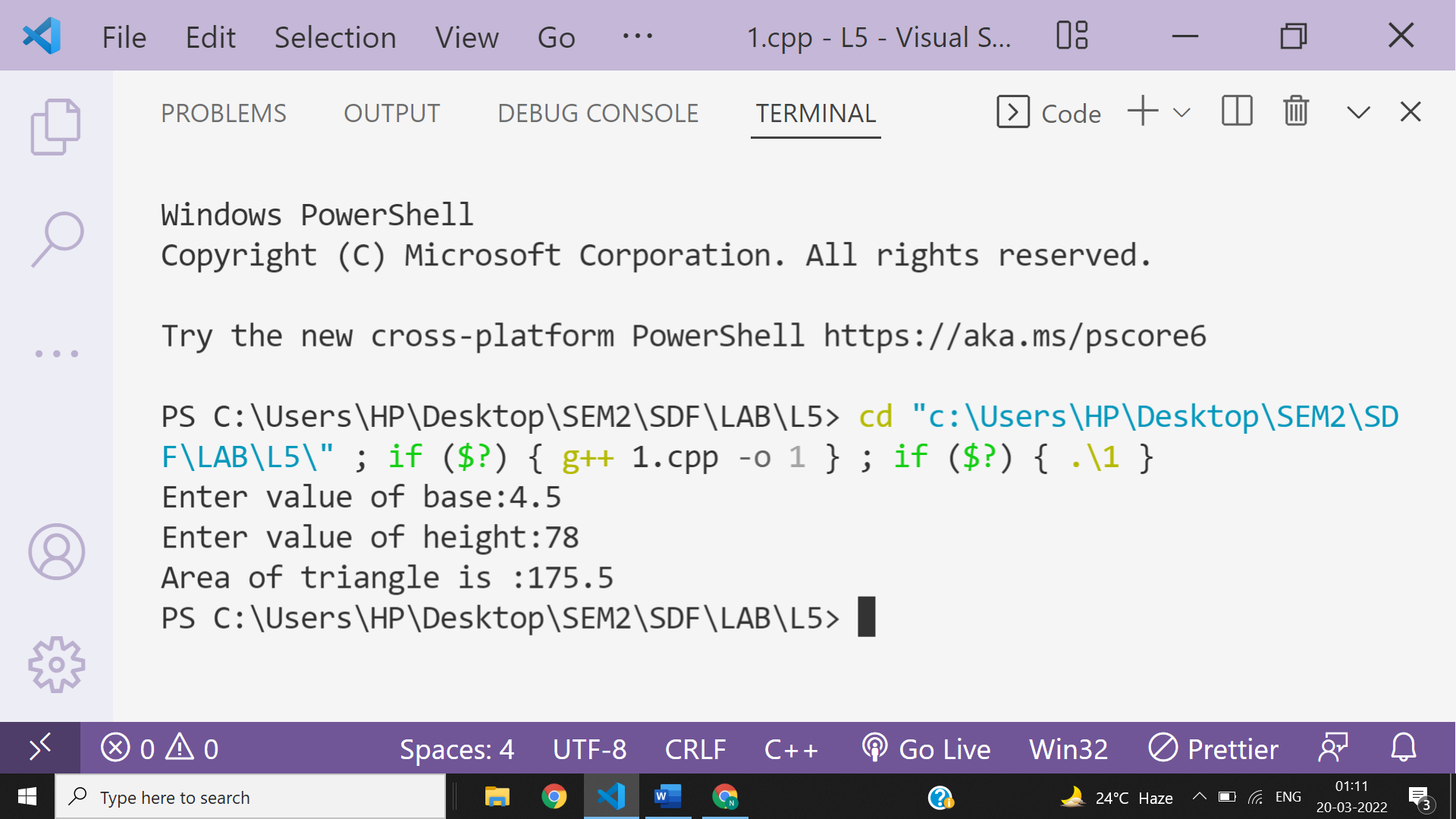
    obj.**getBase**();

    obj.**getHeight**();

    obj.**calArea**();

return 0;

}



**Ans 2-**

#include<iostream>

using namespace **std**;

class **StudentDetails**

{

public:

**string** name;

**string** eno;

void **getDetails**(){

    cout**<<**"enter value of name:";

    cin**>>**name;

    cout**<<**"entr value of eno:";

    cin**>>**eno;

}

};

class **Marks**

{

public:

float marks[5];

void **getMarks**(){

    for(int i=0;i<5;i++){

        cout**<<**"enter value of marks["**<<**i**<<**"] ";

        cin**>>**marks[i];

    }

}

};

class **C**:public **StudentDetails**,public **Marks**

{

public:

float sum=0;

void **calTotal**(){

for(int j=0;j<5;j++){

    sum+=marks[j];

}

cout**<<**"Total="**<<**sum;

}

};

int **main**()

{

**C** obj;

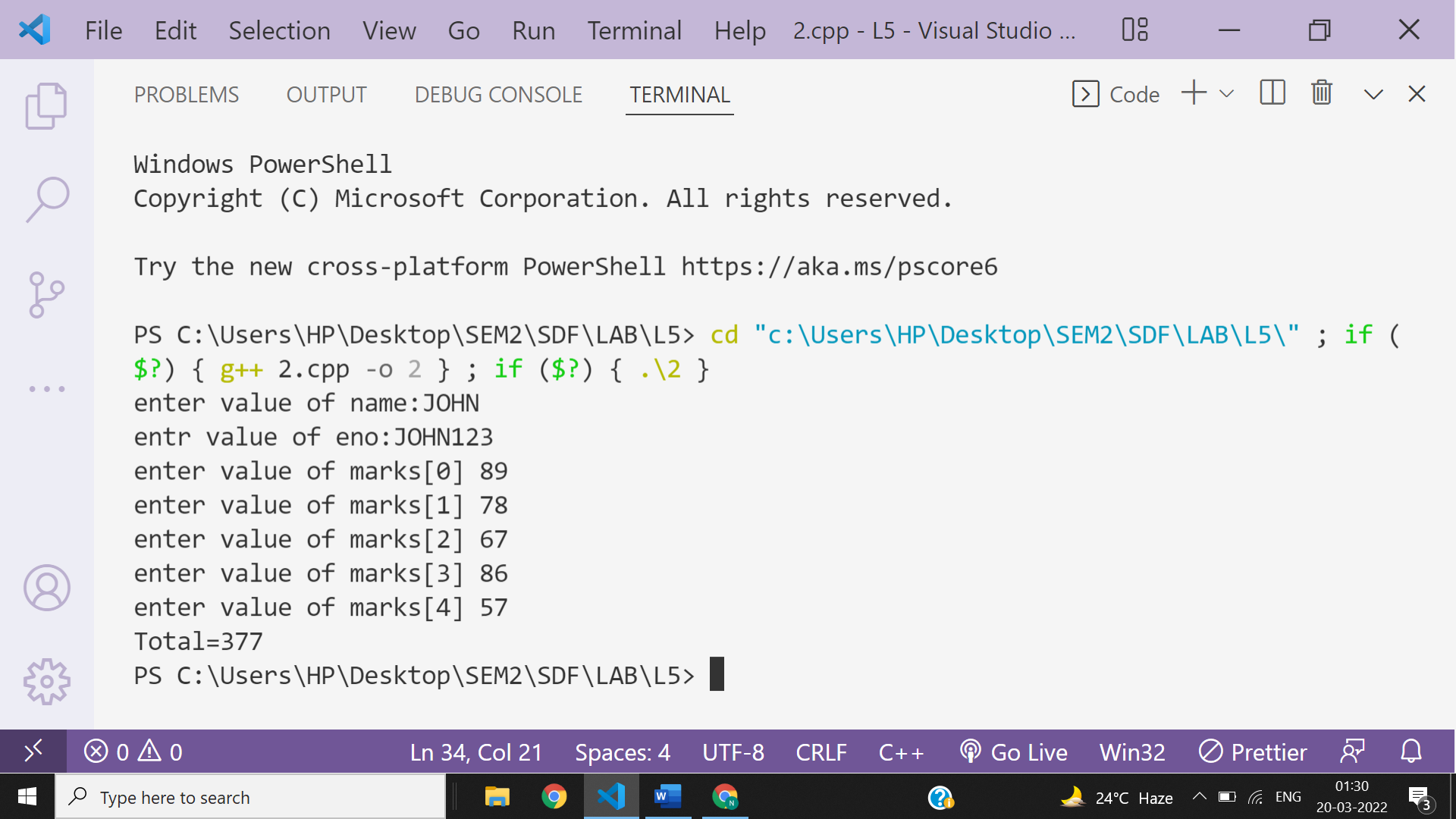
    obj.**getDetails**();

    obj.**getMarks**();

    obj.**calTotal**();

return 0;

}



**Ans 3-**

#include <iostream>

using namespace **std**;

class **base** {

 public:

 char fname[20];

 char surname[20];

public:

virtual void **calculate**()

{

cout **<<** "enter fname:";

cin**>>** fname;

cout **<<** "enter surname";

cin **>>** surname;

}

void **display**()

{

cout **<<** "welcome" **<<** fname **<<** surname**<<endl**;

}

};

class **derived** : public **base** {

public:

void **calculate**()

{

cout **<<** "enter derived\_fname:";

cin**>>**fname;

cout **<<** "enter derived\_ surname";

cin**>>**surname;

}

void **display**()

{

cout **<<** "welcome to derived" **<<** fname **<<** surname**<<endl**;

}

};

int **main**()

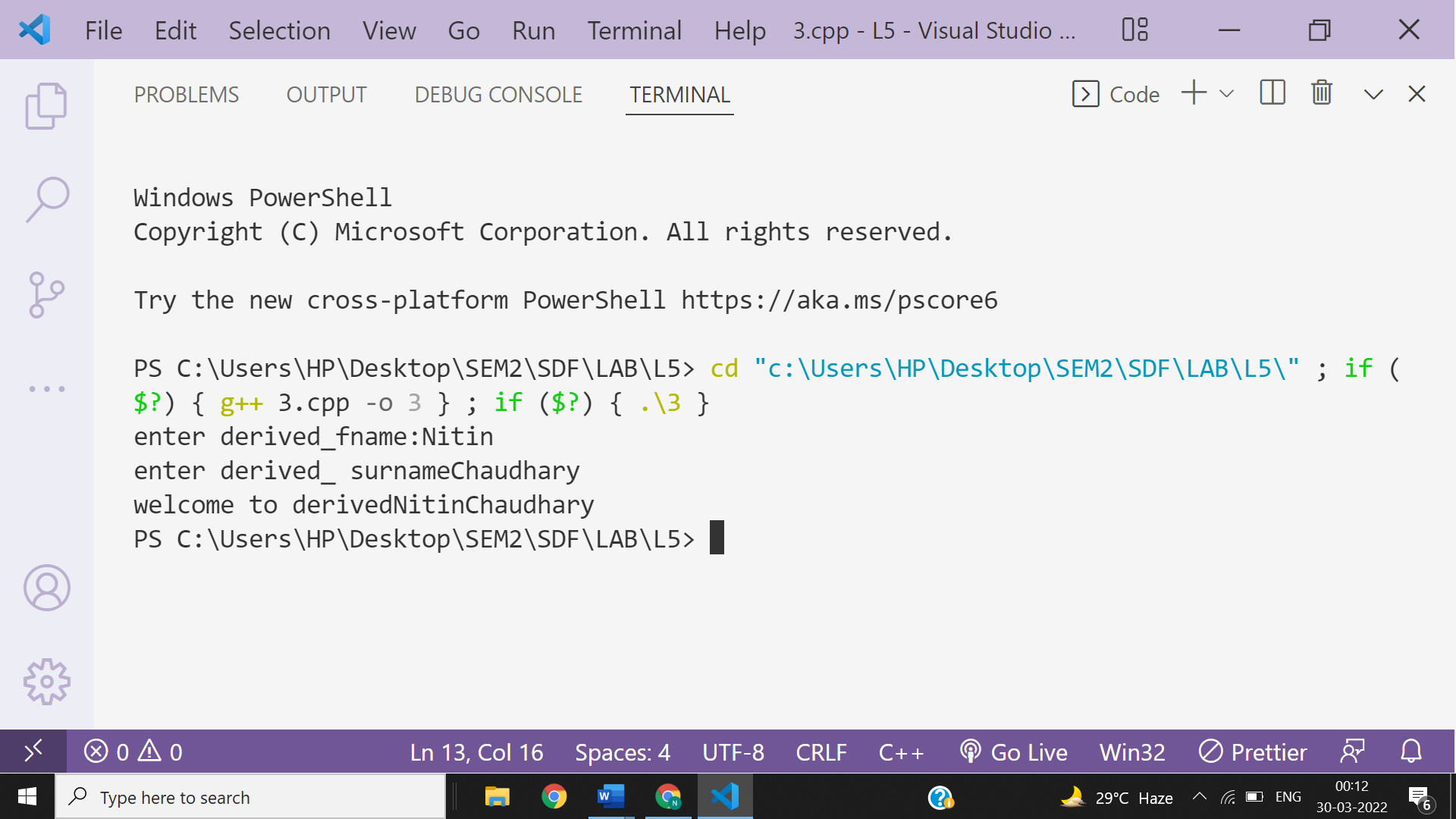
{

**derived** obj;

obj.**calculate**();

obj.**display**();

}



**Ans 4-**

#include <iostream>

using namespace **std**;

class **base\_food\_items** {

 public:

 char item\_name[20];

 int quantity;

 int item\_price;

public:

virtual void **order**()

{

cout **<<** "enter item name:";

cin**>>** item\_name;

cout **<<** "enter quantity";

cin**>>** quantity;

cout **<<** "Item price";

cin **>>** item\_price;

}

void **total\_price**()

{

 cout**<<**"order is: " **<<** item\_name**<<**"\t"**<<**"quantity:"**<<**quantity**<<endl**;

 cout **<<** "total price=" **<<** item\_price\*quantity**<<endl**;

}

};

class **chinese**:public **base\_food\_items**{

public:

void **order**(){

cout **<<** "enter quantity";

cin**>>** quantity;

cout **<<** "Item price";

cin **>>** item\_price;

}

void **total\_price**()

{

 cout**<<**"order is: chinese food" **<<**"\n"**<<**"quantity:"**<<**quantity**<<endl**;

 cout **<<** "\ntotal price=Rs" **<<** item\_price\*quantity**<<endl**;

}

};

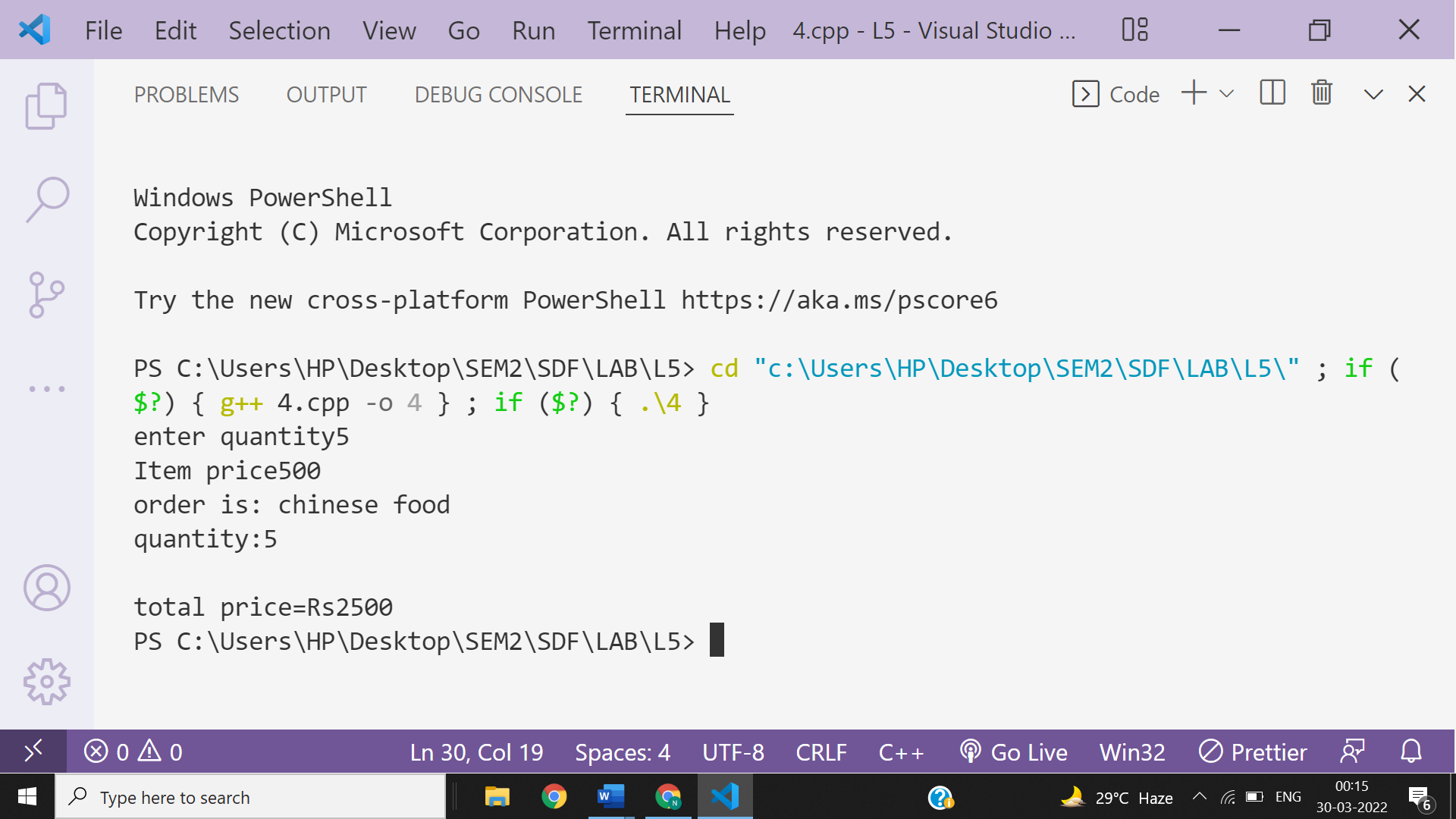
int **main**(){

**chinese** obj;

obj.**order**();

obj.**total\_price**();

}



**Ans 5-**

#include<iostream>

using namespace **std**;

class **Base**{

public:

    virtual void **Display**()=0;

};

class **Derived**:public **Base**{

public:

    void **Display**(){

        cout**<<**"Mandatory to redefine in derived to stop becoming derived class virtual";

    }

    void **Display1**(){

        cout**<<**"This is Display1() method of  Derived Class\n";

    }

    void **Display2**(){

        cout**<<**"This is Display2() method of  Derived Class";

    }

};

int **main**(){

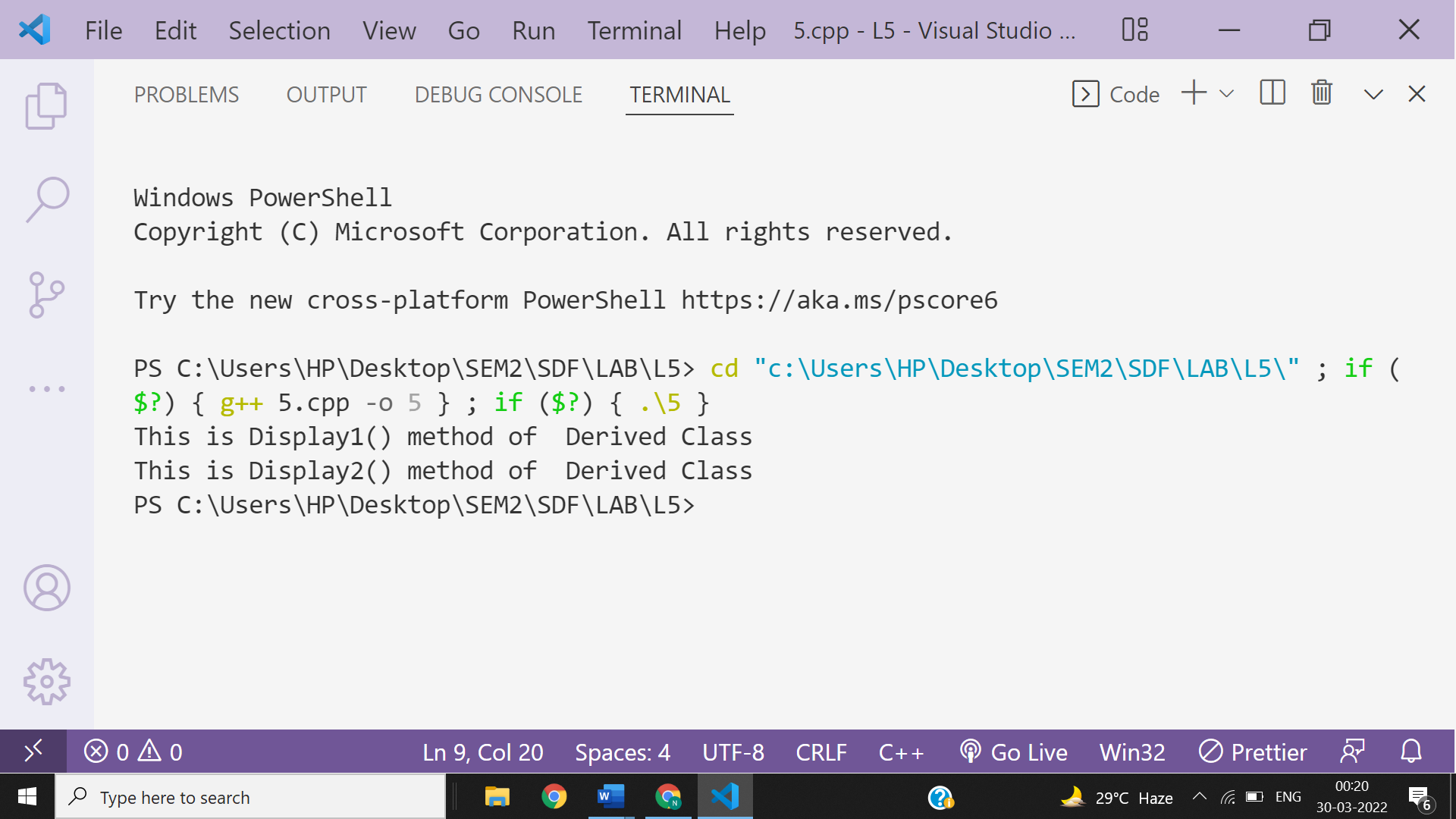
**Derived** obj;

obj.**Display1**();

obj.**Display2**();

return 0;

}



**Ans -6**

#include<iostream>

using namespace **std**;

class **Base**{

public:

    int sum;

**Base**(int i,int j){

        sum=i+j;

        cout**<<**"Sum of two numbers is:"**<<**sum;

    }

    virtual void **Display**()=0;

};

class **Derived**:public **Base**{

public:

void **Display**(){

    cout**<<**"Mandatory to define to stop becoming Derived class abstract";

}

**Derived**(int p,int q):**Base**(p,q){}

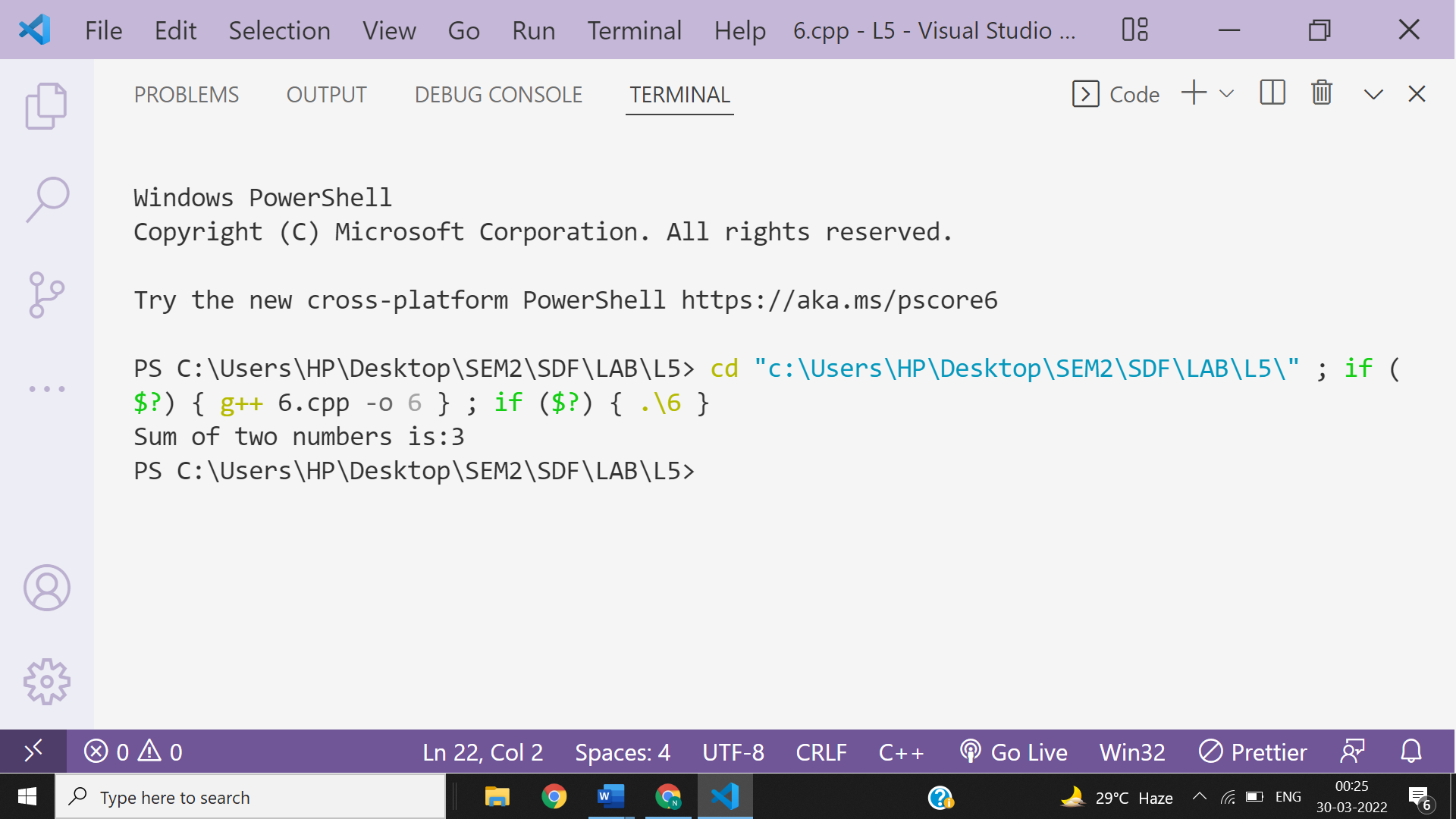
};

int **main**(){

**Derived** **obj**(1,2);

return 0;

}



**Ans 7A-**

#include<iostream>

using namespace **std**;

class **Base** { };

class **Derived**: public **Base** {};

int **main**() {

**Base** \*base\_ptr = new **Derived**;

**Derived** \*derived\_ptr = dynamic\_cast<**Derived**\*>(base\_ptr);

if(derived\_ptr != **NULL**)

cout**<<**"It is working";

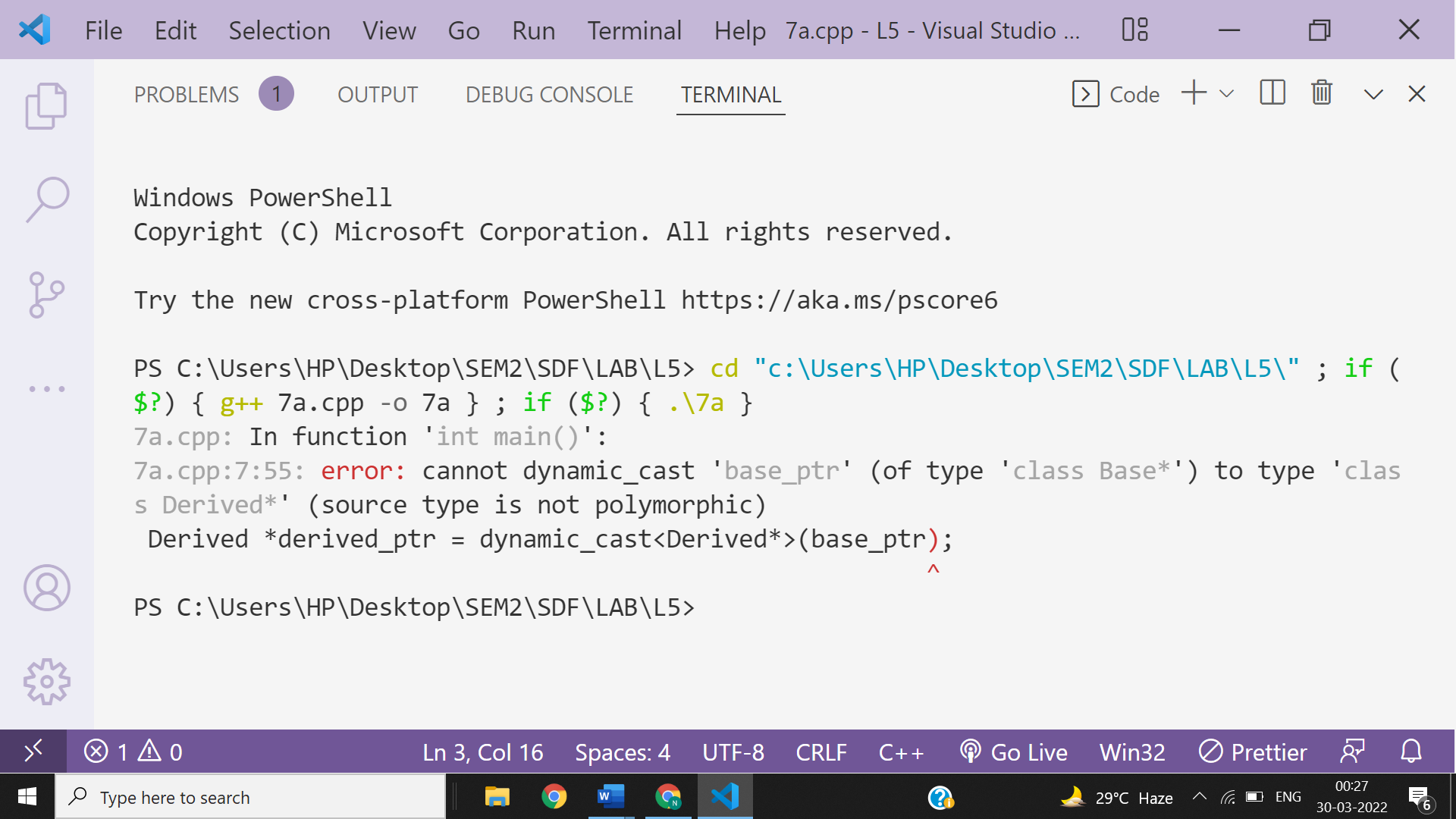
else

cout**<<**"cannot cast Base\* to Derived\*";

return 0;

}

**The above programme will give error as shown below-**



**Explanations-**

**A cast is an operator that converts data from one type to another type. In C++, dynamic casting is mainly used for safe downcasting at run time. To work on dynamic\_cast there must be one virtual function in the base class. A dynamic\_cast works only polymorphic base class because it uses this information to decide safe downcasting.**

**Ans 7B-**

#include<iostream>

using namespace **std**;

class **Base** {

    virtual void **fun**(){

        cout**<<**"It is mandatory to have a virtual function in base class to do dynamic casting";

    }

 };

class **Derived**: public **Base** {};

int **main**() {

**Base** \*base\_ptr = new **Derived**;

**Derived** \*derived\_ptr = dynamic\_cast<**Derived**\*>(base\_ptr);

if(derived\_ptr != **NULL**)

cout**<<**"It is working";

else

cout**<<**"cannot cast Base\* to Derived\*";

return 0;

}

