# EXPERIMENT NUMBER –Practical 5.1

STUDENT’S NAME – Shinde Smita Shahaji

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

TOPIC OF EXPERIMENT –

Write a program that takes information about institute staff information for

1) Teacher code, name, subject and publication

2) Officer code, name and grade

3) Typist code, name, speed and daily wages and displays it using hierarchal inheritance.

# AIM OF THE EXPERIMENT

**Learn how to use classes using C++**

FLOWCHART/ ALGORITHM-

START

Step 1→ Creating a header file for input output stream and define the context.

Step 2 →Creating class name followed by class staff under the inheritance conditions.

Step 3 → Within the class staff defining data members inbound with protected access specifier mode and member function inbound with public access specifier mode. Which can easily to get output out of the class.

Step 4 → Using dummy code of inheritance I created one “class teacher: public staff ” this tell that to complier that we are inheriting class staff in class teachers in public. All the public members of class staff become public members of class teacher.

Step 5 → like class teacher I am creating officer, typist and casual. Under inheritance concept

Step 6 → Create a menu four options in the system.

Step 7 → Initialize the switch statement for each case.

Step 8 → Take input of choice from user.

Step 9 → After input taking process which choice user choose complier ask to user what they want …

1st- create

2nd-display

3rd – go to back main menu.

Step 10 → If stored and asked to show then show the data. According to their construction.

Step 11→ Move to main menu and again ask if choice required.

Step 12→ To exit from the menu press 4.

Stop.

PROGRAM CODE-

#include <iostream>

#include <conio.h>

using namespace std;

class staff

{

protected:

int code;

char name[20];

public:

void getstaff(void)

{

cout<<"\n\nEnter code :-";

cin>>code;

cout<<"Enter name :-";

cin>>name;

}

void dispstaff(void)

{

cout<<"\nNAME :-"<<name;

cout<<"\nCODE :-"<<code;

}

};

class teacher : public staff

{

char sub[20];

char pub[20];

public:

void create(void)

{

getstaff();

cout<<"Enter Subject :-";

cin>>sub;

cout<<"Enter Publication :-";

cin>>pub;

}

void display(void)

{

dispstaff();

cout<<"\nSUBJECT :-"<<sub;

cout<<"\nPUBLICATION:-"<<pub;

}

};

class officer : public staff

{

char grade;

public:

void create(void)

{

getstaff();

cout<<"Enter Grade :-";

cin>>grade;

}

void display(void)

{

dispstaff();

cout<<"\nGRADE :-"<<grade;

}

};

class typist : public staff

{

float speed;

public:

void gettypist(void)

{

getstaff();

cout<<"Enter speed (wpm):-";

cin>>speed;

}

void disptypist(void)

{

dispstaff();

cout<<"\nSPEED :-"<<speed;

}

};

class casual : public typist

{

float dailywages;

public:

void create(void)

{

gettypist();

cout<<"Enter Daily Wages :-";

cin>>dailywages;

}

void display(void)

{

disptypist();

cout<<"\nDAILY WAGES:-"<<dailywages;

}

};

int main()

{

teacher o1t[10];

casual o1c[10];

officer o1o[10];

int choice,i;

char test;

while(1)

{

int count;

start:

cout<<"name- smita shinde. uid -20BCS4643\n ";

cout<<"\n\*\*EDUCATION INSTITUTION DATABASE\*\*\n\n";

cout<<"Choose Category of Information\n";

cout<<"1) Teachers\n";

cout<<"2) Officer\n";

cout<<"3) Typist\n";

cout<<"4) Exit\n";

cout<<"Enter your choice:-";

cin>>choice;

switch(choice)

{

case 1 : while(1)

{

cout<<"\n\*\*TEACHERS INFORMATION\*\*\n\n";

cout<<"\nChoose your choice\n";

cout<<"1) Create\n";

cout<<"2) Display\n";

cout<<"3) Jump to Main Menu\n";

cout<<"Enter your choice:-";

cin>>choice;

switch(choice)

{

case 1 : for(count=0,i=0;i<10;i++)

{

cout<<endl;

o1t[i].create();

count++;

cout<<endl;

cout<<"\n\nAre you Interested in entering data\n";

cout<<"Enter y or n:-";

cin>>test;

if(test=='y' || test=='Y')

continue;

else goto out1;

}

out1:

break;

case 2 : for(i=0;i<count;i++)

{

cout<<endl;

o1t[i].display();

cout<<endl;

}

getch();

break;

case 3 : goto start;

default: cout<<"\nEnter choice is invalid\ntry again\n\n";

}

}

case 2 : while(1)

{

cout<<"\n\*\*OFFICERS INFORMATION\*\*\n\n";

cout<<"\nChoose your choice\n";

cout<<"1) Create\n";

cout<<"2) Display\n";

cout<<"3) Jump to Main Menu\n";

cout<<"Enter your choice:-";

cin>>choice;

switch(choice)

{

case 1 : for(count=0,i=0;i<10;i++)

{

cout<<endl;

o1o[i].create();

count++;

cout<<endl;

cout<<"\n\nAre you Interested in entering data\n";

cout<<"Enter y or n:-";

cin>>test;

if(test=='y' || test=='Y')

continue;

else goto out2;

}

out2:

break;

case 2 : for(i=0;i<count;i++)

{

cout<<endl;

o1o[i].display();

cout<<endl;

}

getch();

break;

case 3 : goto start;

default: cout<<"\nInvalid choice\ntry again\n\n";

}

}

case 3 : while(1)

{

cout<<"\n\*\*TYPIST INFORMATION\*\*\n\n";

cout<<"\nChoose your choice\n";

cout<<"1) Create\n";

cout<<"2) Display\n";

cout<<"3) Jump to Main Menu\n";

cout<<"Enter your choice:-";

cin>>choice;

switch(choice)

{

case 1 : for(count=0,i=0;i<10;i++)

{

cout<<endl;

o1c[i].create();

count++;

cout<<endl;

cout<<"\n\nAre you Interested in entering data\n";

cout<<"Enter y or n:-";

cin>>test;

if(test=='y' || test=='Y')

continue;

else goto out3;

}

out3:

break;

case 2 : for(i=0;i<count;i++)

{

cout<<endl;

o1c[i].display();

cout<<endl;

}

getch();

break;

case 3 : goto start;

default: cout<<"\nInvalid choice\ntry again\n\n";

}

}

case 4 : goto end;

}

}

end:

return 0;

}

ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

(Kindly jot down the compile time errors encountered)

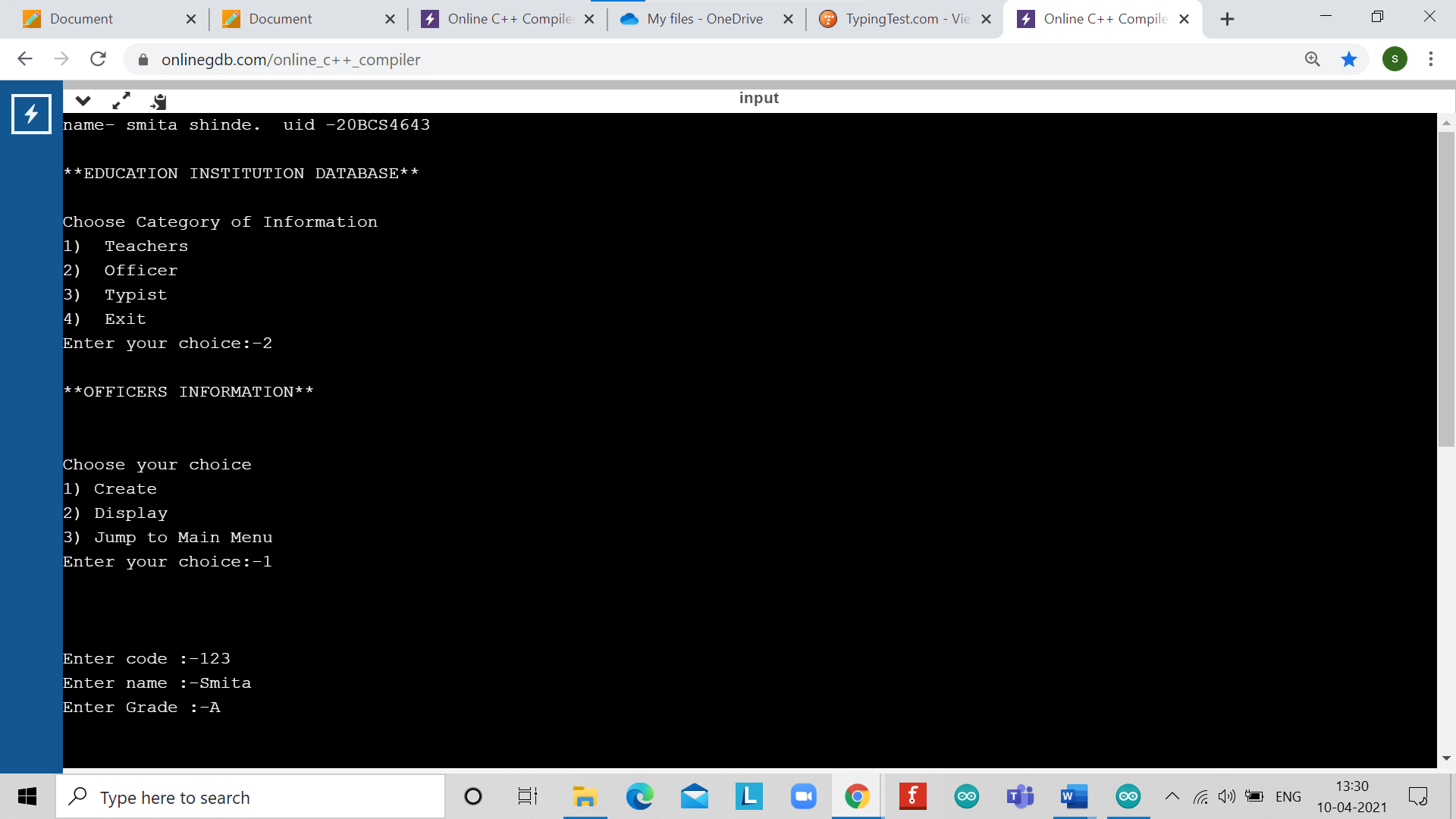
Complier doesn’t have any error but when any wrong input gave to complier program did not stop ..going on and takes to much memory.

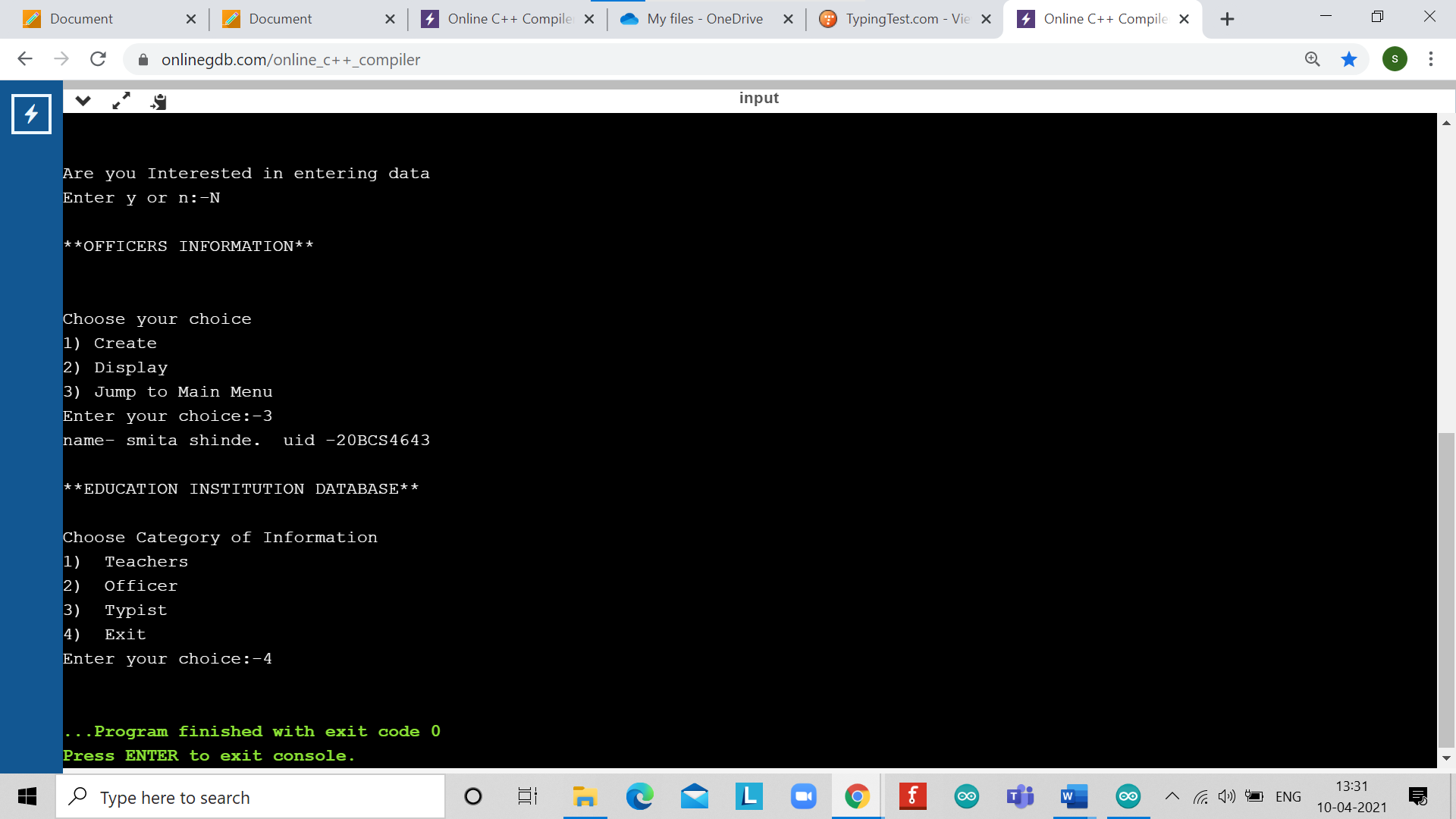
When I used #include <conio.h> 1st time getch(); no declare in function error showing.

PROGRAMS’ EXPLANATION (in brief)

1. We start our program with preprocessor (#) and header file (< iostream >) we have many types of header files but in this program, we used iostream. #include< iostream >, Int main () is a function which work as a container of statements. All the statements are enclosed within the pair of braces { }. “using namespace std” means we use the namespace named std. “std” is an abbreviation for standard. So that means we use all the things with in “std” namespace.
2. Creating class using dummy code of inheritance I created one “class teacher: public staff ” this tell that to complier that we are inheriting class staff in class teachers in public. All the public members of class staff become public members of class teacher. like class teacher I am creating officer, typist and casual. Under inheritance concept.
3. This is a menu driven program where it asks user to choose his designation (teacher, typist, Officer) and ask for various detail like employee code, subject and many other for each designation then stores it. If requested by the user it is being displayed in the screen otherwise, he can again go to main menu and refill any designation data or he can exit from it.

OUTPUT





# EXPERIMENT NUMBER –Practical 5.2

STUDENT’S NAME – Shinde Smita Shahaji

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

TOPIC OF EXPERIMENT –

Create a class student having student uid and getnumber(),putnumber() as member functions to get the values and display it. Derive a class test having marks in different subjects and getmarks() and putmarks() as member functions to get and display the values. Derive another class sports from student class having sports score and getscore(), putscore() as member functions to get and display the values. Derive a class result from test

and sports class and define a function display() to calculate total marks. Implement it with the object of result class. If it gives any error, resolve it by adding the required functionality.

AIM OF THE EXPERIMENT

**Learn how to use classes using C++**

FLOWCHART/ ALGORITHM

Start.

Step 1→ Creating a header file for input output stream and define the context.

# Step 2 → After that I create a class student and declare data member int UID;

Step 3 → Declare and define the functions getnumber() and putnumber() within public access specifier.

Step 4→ Create the derived class test virtually derived from the base class student.

Step 5→ Declare and define the function getmarks() and putmarks().

Step 6→ Create the derived class sports virtually derived from the base class student.

Step 7→ Declare and define the function getscore() and putscore().

Step 8→ Create the derived class result derived from the class test and sports.

Step 9→ Declare and define the function display () to calculate the total.

Step 10→ Create the derived class object obj1. For calling member functions.

Step 11→ Call the function get number(),getmarks(),getscore() and display() using dot(.) operator .

# Step 12→ End the program by returning UID of the student and their marks of maths and science and sport score and total of maths and science marks what was user giving input to program .

# Stop.

PROGRAM CODE

#include<iostream>

#include<conio.h>

using namespace std;

class student

{

int UID;

public:

void getnumber()

{

cout<<"==| smita shinde |==\n";

cout << "Enter UID No:";

cin>>UID;

}

void putnumber() {

cout << "\n\n\tUID:" << UID << "\n";

}

};

class test : virtual public student

{

public:

int maths, science;

void getmarks()

{

cout << "Enter Marks\n";

cout << "maths:";

cin>>maths;

cout << "science:";

cin>>science;

}

void putmarks()

{

cout << "\tMarks Obtained\n";

cout << "\n\tmaths:" << maths;

cout << "\n\tscience:" << science;

}

};

class sports : public virtual student {

public:

int score;

void getscore() {

cout << "Enter Sports score:";

cin>>score;

}

void putscore() {

cout << "\n\tSports Score is:" << score;

}

};

class result : public test, public sports

{

int total;

public:

void display() {

total = maths+ science + score;

putnumber();

putmarks();

putscore();

cout << "\n\tTotal Score:" << total;

}

};

int main() {

result obj1;

obj1.getnumber();

obj1.getmarks();

obj1.getscore();

obj1.display();

return 0;

}

ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

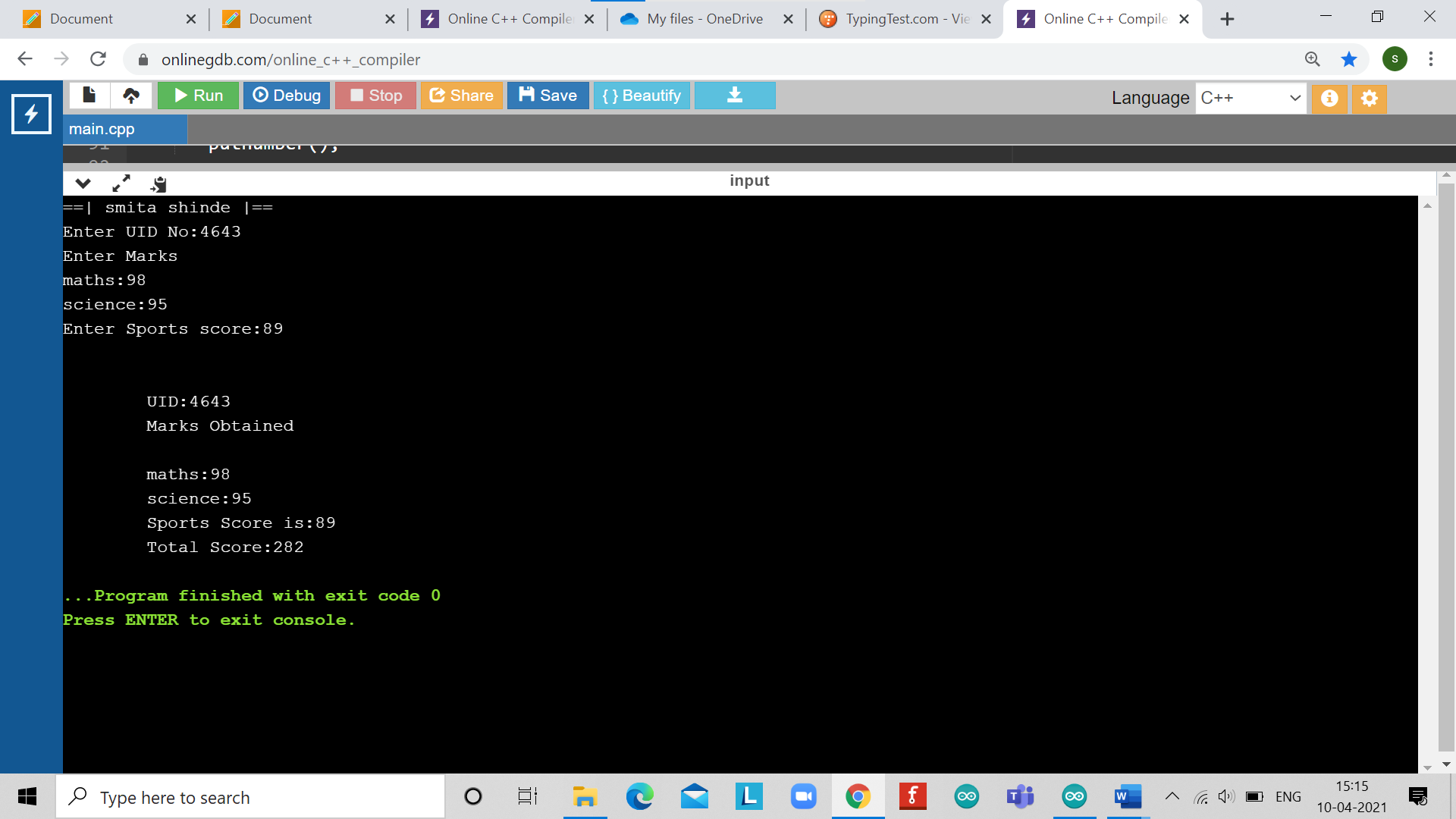
(Kindly jot down the compile time errors encountered)

1. Semicolons was missing.
2. Parenthesis(brackets) was missing.
3. Double quotes were missing in string.

PROGRAMS’ EXPLANATION (in brief)

1. We start our program with preprocessor (#) and header file (< iostream >) we have many types of header files but in this program, we used iostream. #include< iostream >, Int main () is a function which work as a container of statements. All the statements are enclosed within the pair of braces { }. “using namespace std” means we use the namespace named std. “std” is an abbreviation for standard. So that means we use all the things with in “std” namespace.
2. In this program we create a class student having student uid and getnumber(),putnumber() as member functions to get the values and display it. Derive a class test having marks in different subjects and getmarks() and putmarks() as member functions to get and display the values. Derive another class sports from student class having sports score and getscore(), putscore() as member functions to get and display the values. Derive a class result from test and sports class and define a function display() to calculate total marks. Then using the object calling we call each of its object in the final class to show the output in the output screen.

OUTPUT



# EXPERIMENT NUMBER –Practical 5.3

STUDENT’S NAME – Shinde Smita Shahaji

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

TOPIC OF EXPERIMENT –

WAP to illustrate how the constructors are implemented and the order in which they are called when the classes are inherited. Use three classes named alpha, beta, gamma such that alpha, beta are base class and gamma is derived class inheriting alpha &beta. Pass four variable to gamma class object which will further send one integer variable to alpha(),one float type variable to beta().Show the order of execution by invoking constructor of derived class.

AIM OF THE EXPERIMENT

**Learn how to use classes using C++**

FLOWCHART/ ALGORITHM

Start.

Step 1→ Creating a header file for input output stream and define the context.

# Step 2 → After that I create a class alpha and declare data member int x;

Step 3 → Declare and define the functions void show\_x() within public access specifier mode

Step 4→ 2nd class is created name follow as class beta and declare data members.

Step 5→ Declare and define the function void show\_y() within public access specifier mode.

Step 6→ Using dummy code of inheritance I created one “class gamma: public alpha public beta ” this tell that to complier that we are inheriting class alpha beta in class gamma in public. All the public members of class gamma become public members of class alpha and beta.

Step 7→ Declare and define the function of class gamma void show\_mn() is the member function and int a,float b,int c,int d; this are data members.

Step 8→ Create the derived int main() to calling all the member functions.

Using object obj.1

Step 9→ Pass 4 variables to gamma each of different data type

Step 10→ Call the function void show\_x(), void show\_y() , void show\_mn() using dot(.) operator .

Step 11→Display the data in the order of execution.

Stop

PROGRAM CODE

#include<iostream>

using namespace std;

class alpha

{

int x;

public:

alpha(int i)

{

x=i;

cout<<"== | smita shinde |==";

cout<<"alpha initialized\n";

}

void show\_x()

{

cout<<"x="<<x<<"\n";

}

};

class beta

{

float y;

public:

beta(float j)

{

y=j;

cout<<"beta initialized\n";

}

void show\_y()

{

cout<<"y="<<y<<"\n";

}

};

class gamma : public alpha,public beta

{

int m,n;

public:

gamma(int a,float b,int c,int d): alpha(a),beta(b)

{

m=c,n=d;

cout<<"gamma initialized\n";

}

void show\_mn()

{

cout<<"m="<<m<<"\n";

cout<<"n="<<n<<"\n";

}

};

int main()

{

gamma obj1(20,19.75,50,90);

obj1.show\_x();

obj1.show\_y();

obj1.show\_mn();

return 0;

}

ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

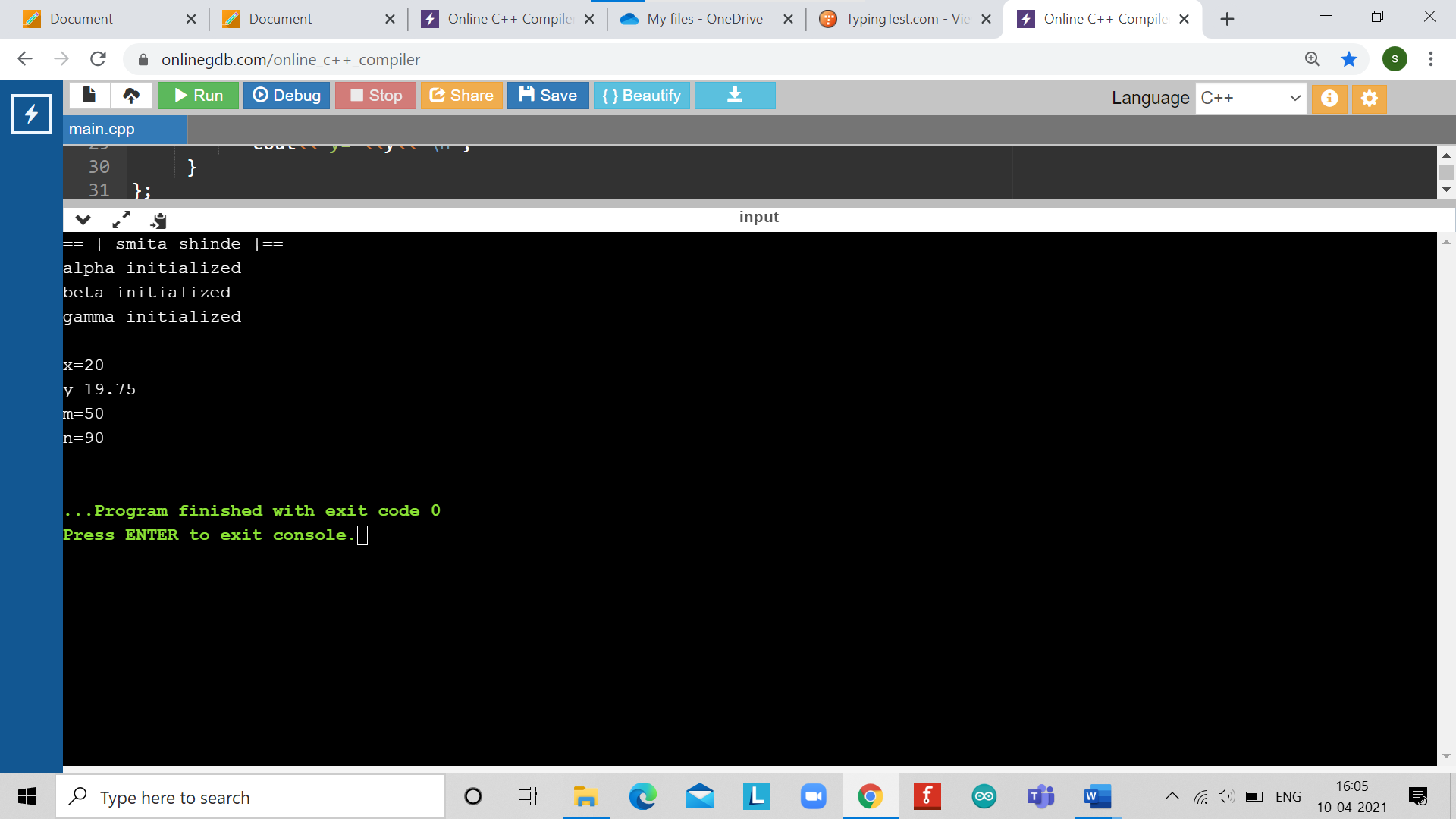
(Kindly jot down the compile time errors encountered)

NO error

PROGRAMS’ EXPLANATION (in brief)

1. We start our program with preprocessor (#) and header file (< iostream >) we have many types of header files but in this program, we used iostream. #include< iostream >, Int main () is a function which work as a container of statements. All the statements are enclosed within the pair of braces { }. “using namespace std” means we use the namespace named std. “std” is an abbreviation for standard. So that means we use all the things with in “std” namespace.
2. In this program we creating Use three classes named alpha, beta, gamma such that alpha, beta are base class and gamma is derived class inheriting alpha &beta. Pass four variable to gamma class object which will further send one integer variable to alpha(),one float type variable to beta().Show the order of execution by invoking constructor of derived class.

output



LEARNING OUTCOMES

|  |
| --- |
| * Identify situations where computational methods would be useful. |
| * Approach the programming tasks using techniques learnt and write pseudo-code. |
| * Choose the right data representation formats based on the requirements of the problem. |
| * Use the comparisons and limitations of the various programming constructs and choose the right one for the task. |

EVALUATION COLUMN (To be filled by concerned faculty only)

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Parameters** | **Maximum**  **Marks** | **Marks**  **Obtained** |
| 1. | Worksheet Completion including writing learning objective/ Outcome | 10 |  |
| 2. | Post Lab Quiz Result | 5 |  |
| 3. | Student engagement in Simulation/ Performance/ Pre Lab Questions | 5 |  |
| 4. | Total Marks | 20 |  |