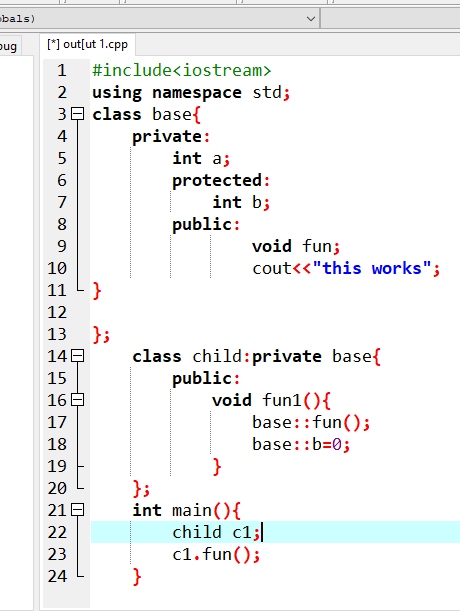
|  |
| --- |
| Assignment of MID |
|  |
| **Name : Fatima Fiaz SAP ID : 45470 BS CS 2nd sem 2A Subject : OOP** |

[Pick the date]

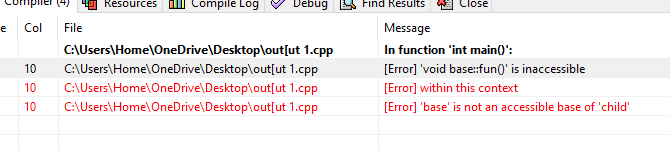
**Assignment of Mid Term Exam**

* **Question 1: write the output of following code snippets.**

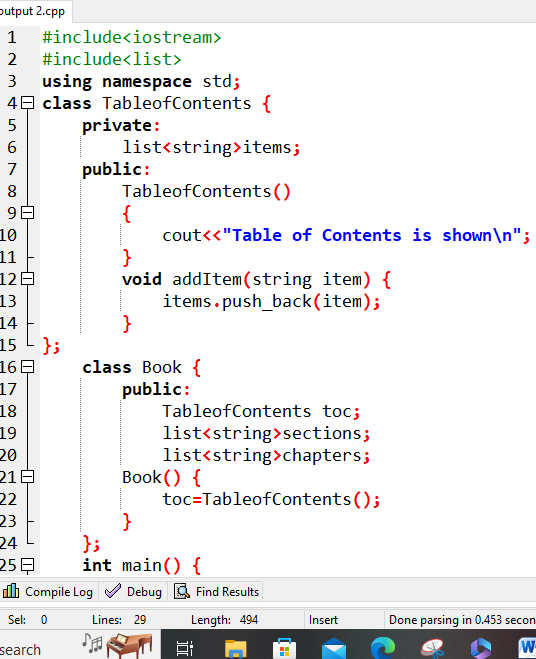
Program 1:



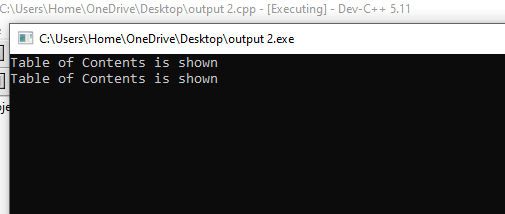
Output = errors



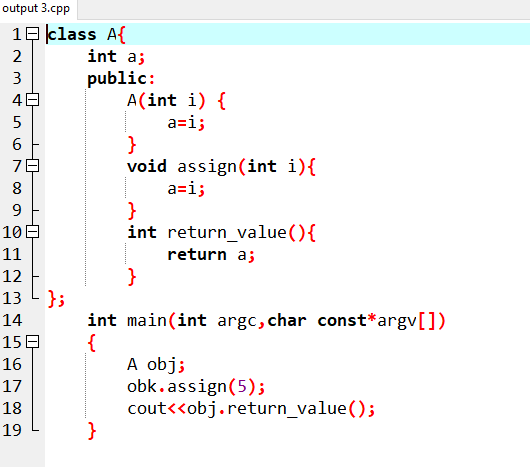
Program 2:



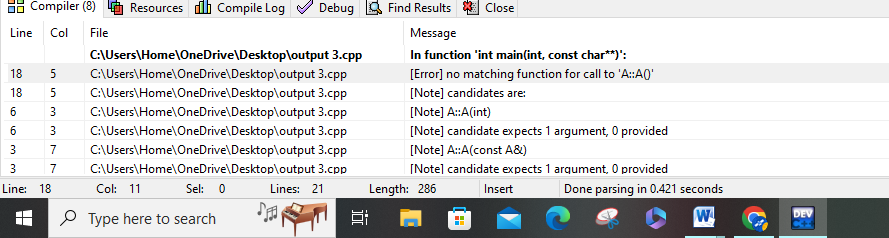
Output =



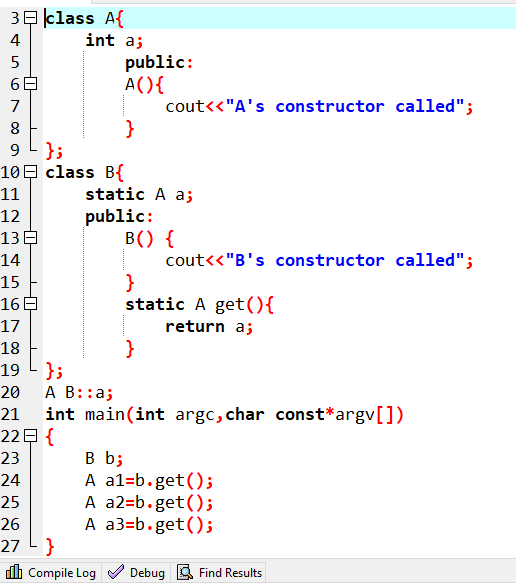
Program 3:



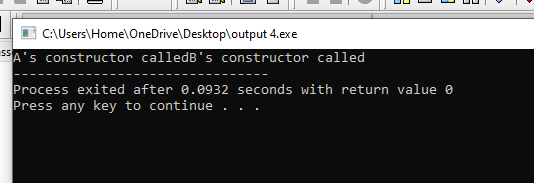
Output = Errors



Program 4:



Output = A’s constructor will call 1 time.



* **Q2 . write answers of the questions.**

**1. What is the purpose of access modifiers OOP language?**

**ANS**. In object-oriented languages, access modifiers (or access specifiers) are keywords that control the accessibility of classes, methods, and other members. Access modifiers are a type of programming language syntax that makes it easier to encapsulate components. Using the access modifiers we can set the scope or accessibility of these classes, methods, constructors, and other members.

**2. If we want to access the private members of a class in the child class what do we need to change?**

**ANS**. To access these private members in the child class, you must use setter and getter methods. because setter and getter allow you to both read from and write to private members of the superclass .

* **Question 3.Determine the accessibility of functions and data members in the following scenarios:**

|  |  |  |
| --- | --- | --- |
| **Scenarios** | **Accessible** | **Not Accessible** |
| **A private data member is declared in a class in accessible by its object in the main function.** |  | Not Accessible |
| **A Protected function defined in parent class by the functions of the child class.** | Accessible |  |
| **A public data member of the parent class by the object of child class** | Accessible |  |

* **Ques 4 . Identify the errors.**

**CODE:**

class B1{

public:

int i;

int j;

void g(int){

}

};

class B2{

public :

int j;

void g(){

}

};

class D: public B1, public B2 {

public:

int i;

};

int main()

{

D dobj;

D \*dptr=&dobj;

dptr->i=5;

dptr->i=10;

}

**ERRORS:**

* i; in line 4. i was not declared
* class D: public B1; class public B2
* the inheritance should be on a seprate line by using ","
* dobj.g(); B2 has an empty body so calling a function in main
* **Question 5 :**

**You have to develop a game that has multiple characters. These characters share some common properties like id, name, maximum power and strength. There are other properties as well that they have their own like Doremon has properties like a list of names gadgets and the name of partner, Benten has the watch Name, a list of names powers and total charge of the watch. There are also some common actions that they can perform like walk, jump and eat. Doremon can show Gadgets, launch attack and fly. Benten can perform the actions like rotate watch, fight and drive.**

**Implement the game using Inheritance in C++.**

Program

#include <iostream>

#include <string>

using namespace std;

class Character {

protected:

int id;

string name;

int max\_power;

int strength;

public:

Character(int id, string name, int max\_power, int strength) {

this->id = id;

this->name = name;

this->max\_power = max\_power;

this->strength = strength;

}

virtual void walk() {

cout << name << " is walking." << endl;

}

virtual void jump() {

cout << name << " is jumping." << endl;

}

virtual void eat() {

cout << name << " is eating." << endl;

}

};

class Doremon : public Character {

private:

int num\_gadgets;

string partner\_name;

public:

Doremon(int id, string name, int max\_power, int strength, int num\_gadgets, string partner\_name) : Character(id, name, max\_power, strength) {

this->num\_gadgets = num\_gadgets;

this->partner\_name = partner\_name;

}

void show\_gadgets() {

cout << name << " has " << num\_gadgets << " gadgets." << endl;

}

void launch\_attack() {

cout << name << " is launching an attack." << endl;

}

void fly() {

cout << name << " is flying." << endl;

}

};

class Benten : public Character {

private:

string watch\_name;

int num\_powers;

int total\_charge;

public:

Benten(int id, string name, int max\_power, int strength, string watch\_name, int num\_powers, int total\_charge) : Character(id, name, max\_power, strength) {

this->watch\_name = watch\_name;

this->num\_powers = num\_powers;

this->total\_charge = total\_charge;

}

void rotate\_watch() {

cout << name << " is rotating the watch." << endl;

}

void fight() {

cout << name << " is fighting." << endl;

}

void drive() {

cout << name << " is driving." << endl;

}

};

int main() {

Doremon doremon(1, "Doremon", 100, 50, 10, "Nobita");

doremon.walk();

doremon.jump();

doremon.eat();

doremon.show\_gadgets();

doremon.launch\_attack();

doremon.fly();

Benten benten(2, "Benten", 150, 75, "Omnitrix", 20, 50);

benten.walk();

benten.jump();

benten.eat();

benten.rotate\_watch();

benten.fight();

benten.drive();

return 0;

}

* **Output :**

