

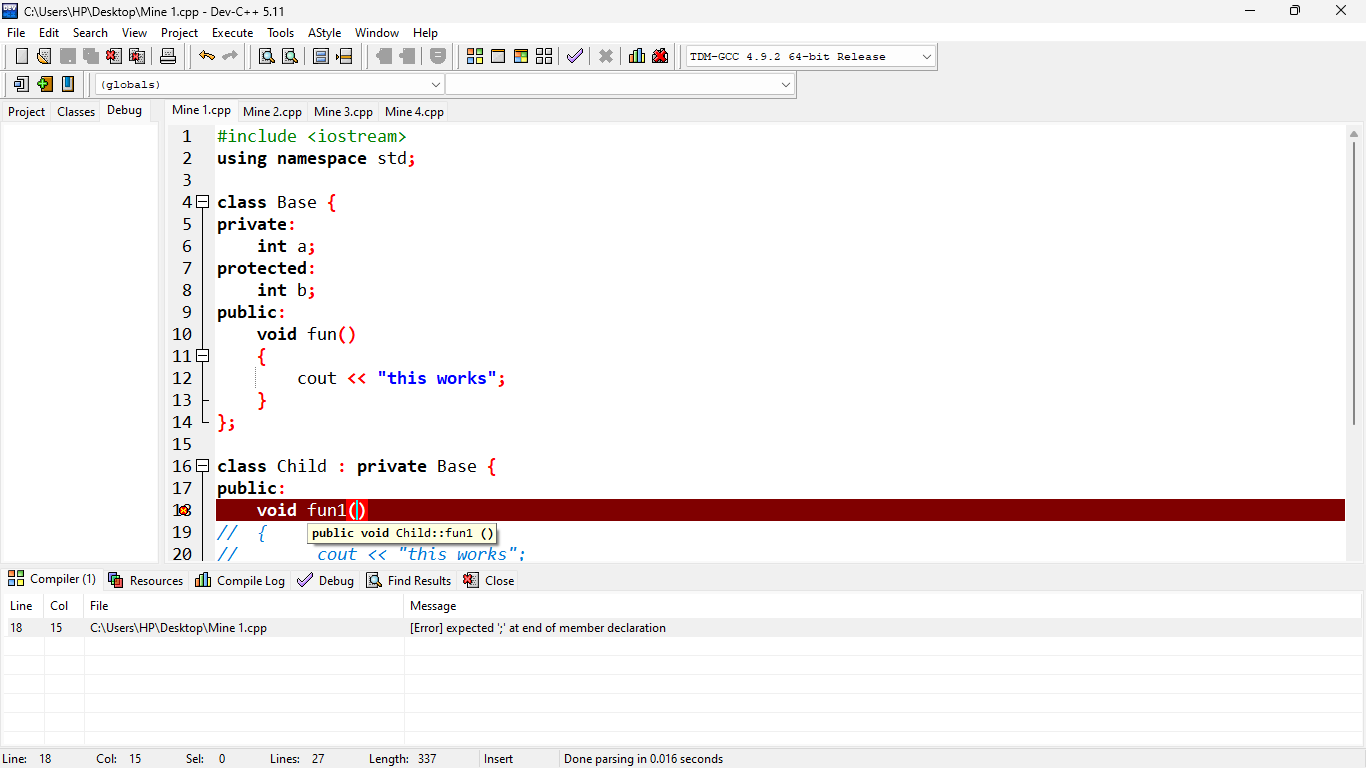
**ZOHAIB MEHRAN**

**46728**

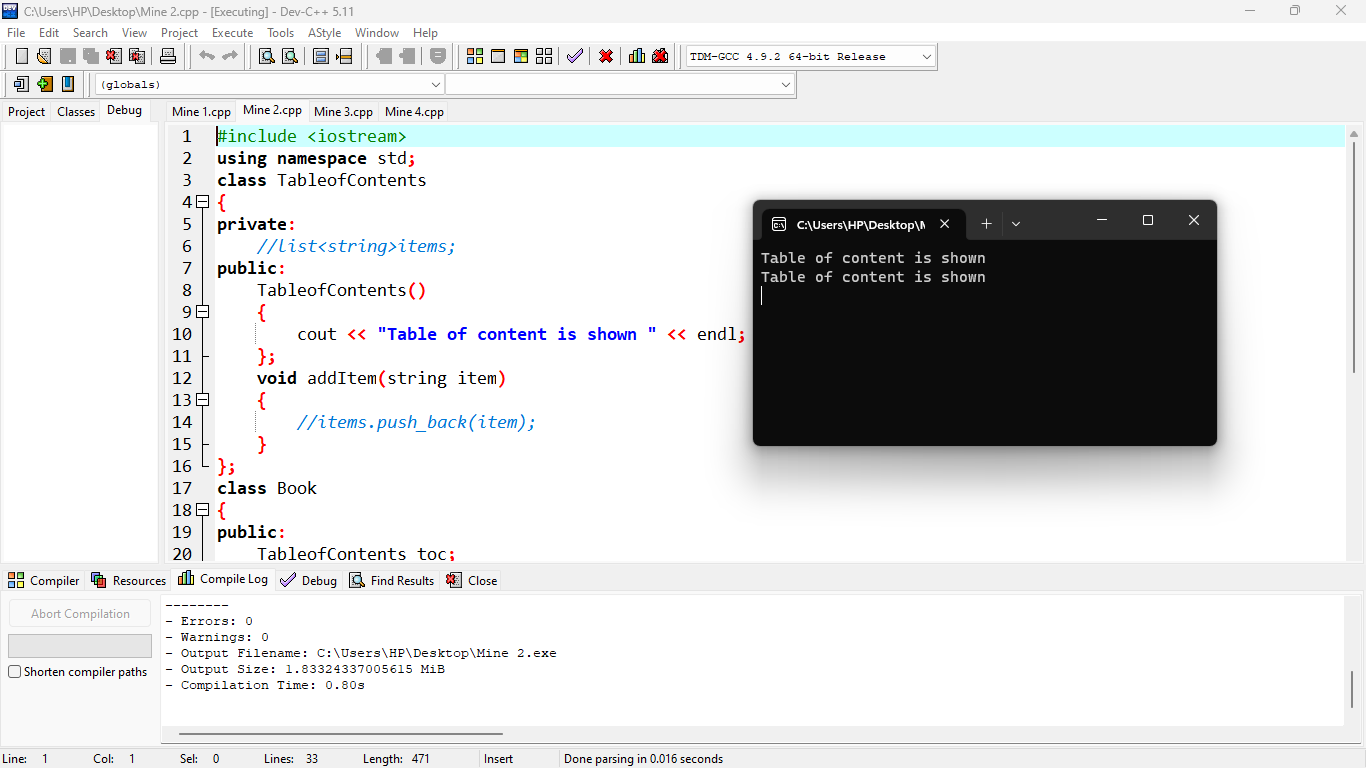
**OOP ASSIGNMENT (MIDS PAPER)**

**Question 1:**

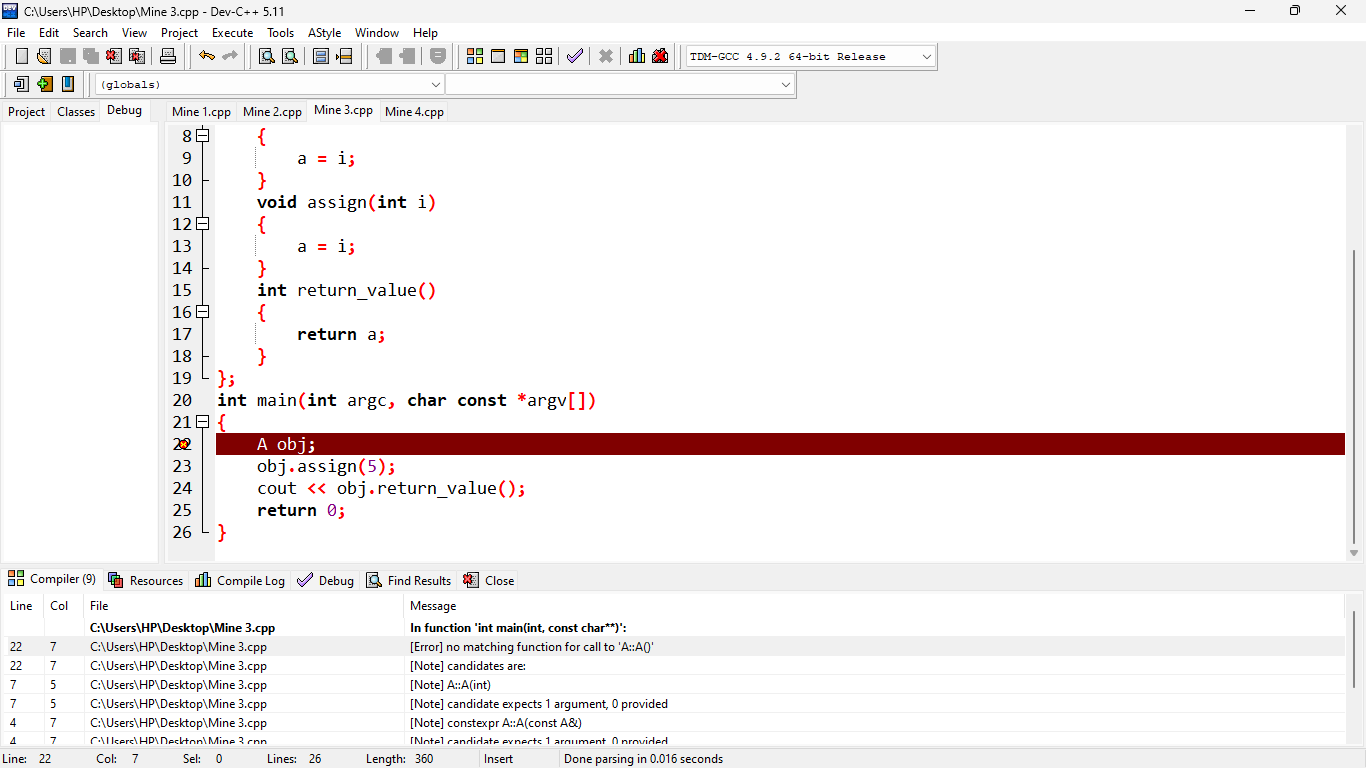
* **Answer:** ERROR

****

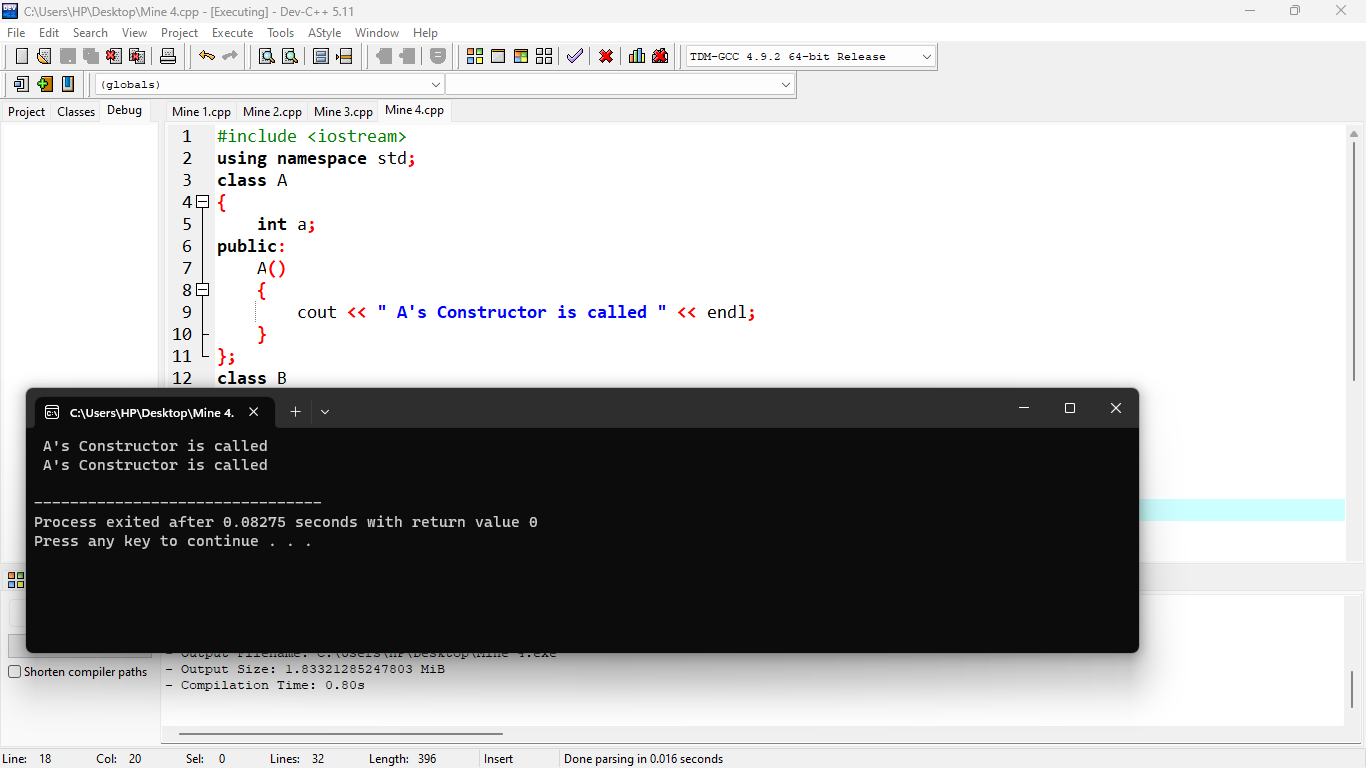
* **Answer** : Table Of Content is shown

****

* **Answer** : Error

****

* **Answer :** A’s constructor called

****

**Question 2:**

* **What is the purpose of access modifiers in OOP languages?**

**Public::**The public access modifier is the direct opposite of the private access modifier. A class, method or variable can be declared as public and it means that it is accessible from any class. Public access modifier can be likened to a public school where anyone can seek admission and be admitted.

**Private:** The private access modifier makes the class member accessible only within the same class where it is declared. Private members cannot be accessed from outside the class, including from any subclasses or other parts of the program. This helps to ensure that the internal state of the class is not modified in unexpected ways

**Protected:** It is a keyword. This access modifier is used to access the methods or data members of a class within the same package as well as outside the package but only through inheritance. The protected access modifier has more accessibility than private and defaults access modifiers.

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

In object-oriented programming, private members of a class are not visible outside the class, including in child classes. However, there are certain cases where you might want to access private members of a class in a child class. In order to achieve this, you need to use the concept of inheritance and access modifiers.

To access private members of a parent class in a child class, you can change the access modifier of those members to protected. The protected access modifier allows the members to be accessed by the child class, as well as any other class that extends the parent class.

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

1. Not Accessible
2. Accessible
3. Accessible

* **QUESTION 3:**

**Errors :**

There are 5 errors in the given code:

1. Line no 3: i; -> int i;
2. Line no 5: void g() has no definition.
3. Line no 12: The inheritance of class B2 is incorrect.
4. Line no 17: D obj is not declared in any class.
5. Line no 18: Using pointer without declaration.

* **Question 4 :**

**CODE :**

#include <iostream>

#include <string>

using namespace std;

class Character {

public:

int id;

string name;

int max\_power;

int strength;

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

this->id = id;

this->name = name;

this->max\_power = max\_power;

this->strength = strength;

}

void walk() {

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

}

void jump() {

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

}

void eat() {

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

}

};

class Doremon : public Character {

public:

string gadgets[3];

string partner\_name;

Doremon(int id, string name, int max\_power, int strength, string gadgets[3], string partner\_name)

: Character(id, name, max\_power, strength) {

for (int i = 0; i < 3; i++) {

this->gadgets[i] = gadgets[i];

}

this->partner\_name = partner\_name;

}

void showGadgets() {

cout << "Gadgets: ";

for (int i = 0; i < 3; i++) {

cout << gadgets[i] << ", ";

}

cout << endl;

}

void launchAttack() {

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

}

void fly() {

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

}

};

class Benten : public Character {

public:

string watch\_name;

string powers[3];

int watch\_charge;

Benten(int id, string name, int max\_power, int strength, string watch\_name, string powers[3], int watch\_charge)

: Character(id, name, max\_power, strength) {

this->watch\_name = watch\_name;

for (int i = 0; i < 3; i++) {

this->powers[i] = powers[i];

}

this->watch\_charge = watch\_charge;

}

void rotateWatch() {

cout << "Rotating the " << watch\_name << " watch." << endl;

}

void fight() {

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

}

void drive() {

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

}

};

int main() {

string gadgets[3] = {"Anywhere Door", "Takecopter", "Small Light"};

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

string powers[3] = {"Fireball", "Thunderbolt", "Big Bang"};

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

doremon.walk();

doremon.showGadgets();

doremon.launchAttack();

doremon.fly();

benten.jump();

benten.rotateWatch();

benten.fight();

benten.drive();

return 0;

}