

**NAME: MUHAMMAD IBRAHIM**

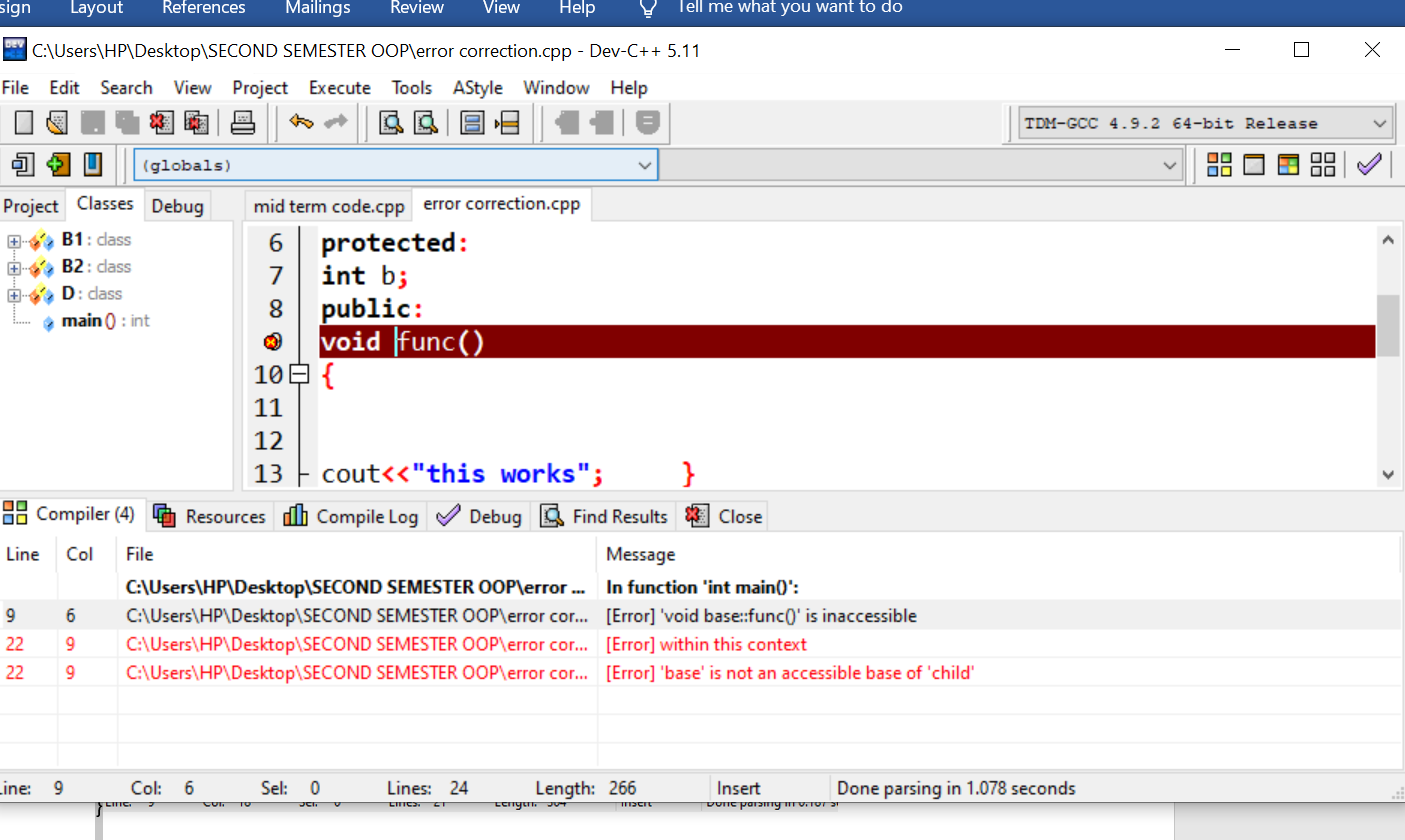
**SAPID:46935**

**OBJECT ORIENTED PROGRAMMING:**

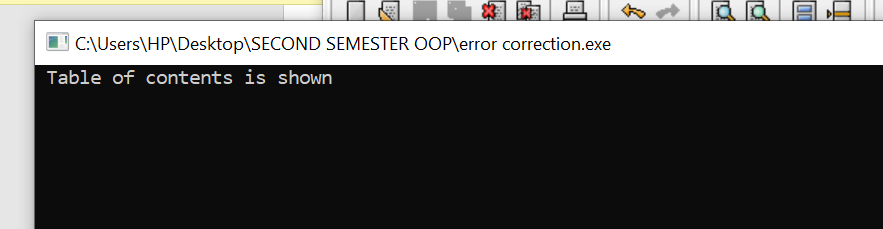
**ASSIGNMENT (MID TERM PAPER)**

**Question 1: Write the outputs of following code snippets.**

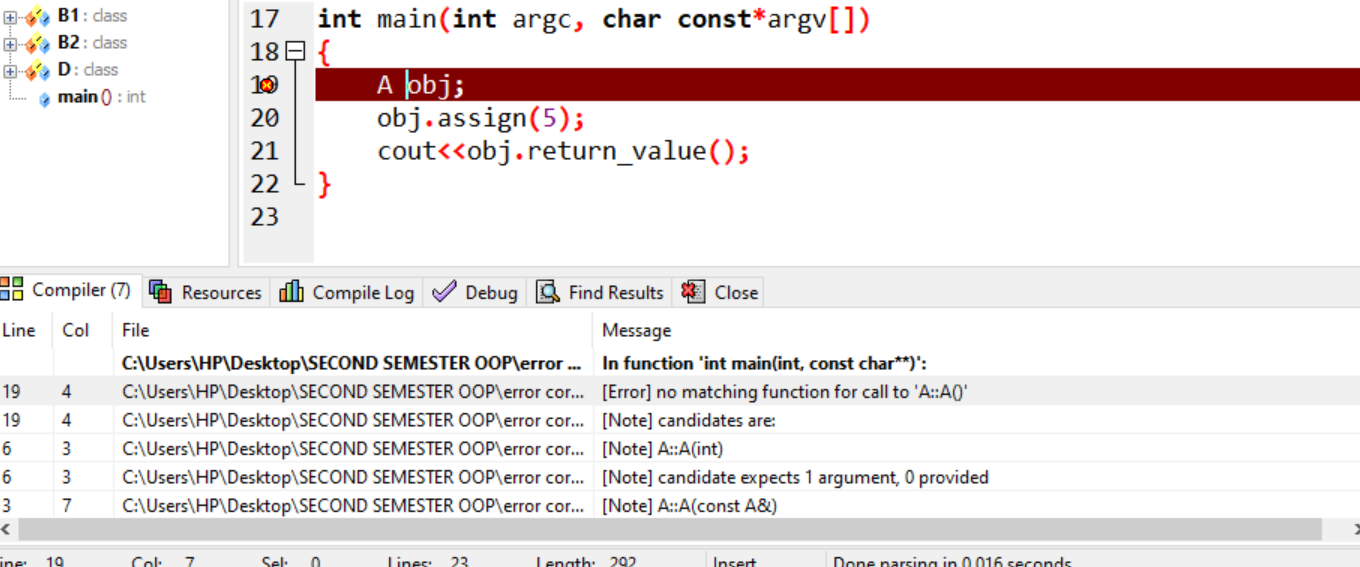
1. **OUTPUT : ERROR**

****

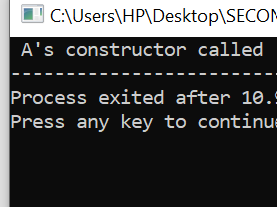
1. **OUTPUT : Table Of Content is shown**

****

1. **OUTPUT : Error**

****

1. **OUTPUT : A’s constructor called**

****

**Question 2: Give Short Questions of the following Questions :**

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

**Public:** The public access modifier makes the class member accessible from anywhere in the program, including outside the class. Public members can be accessed and modified by any part of the program, which can make them useful for creating a public interface for a class.

**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:** The protected access modifier makes the class member accessible within the same class and any subclasses derived from it. Protected members cannot be accessed from outside the class hierarchy. This helps to ensure that the class can be extended safely without exposing its internal implementation details.

1. **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.

1. **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** |  |

**Q3. 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;

}

**ERRROR:**

**1-LINE NO 3:**

**CORRECT LINE OF CODE:**

**Int i;**

**2-LINE NO 12:**

**CORRECT LINE OF CODE:**

**Class D:public B1,public B2**

**//; TO ,**

**3-LINE NO 12:**

**CORRECT LINE OF CODE:**

**Class D:public B1,public B2**

**//REMOVE THE WORD CLASS**

**4-LINE NO 21:**

**Dobj.g();**

**Question 4 :**

**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++.**

**CODE :**

#include<iostream>

using namespace std;

class game{

public:

int id;

string name;

int maxpower;

int strength;

};

class doremon:public game{

public:

string gadgets;

string partners;

void display(){

cout<<"name of gadgtes"<<endl;

cin>>gadgets;

cout<<"name of partners"<<endl;

cin>>partners;

}

void action(){

cout<<"they can jump"<<endl;

cout<<"they can eat:"<<endl;

cout<<"they can walk:"<<endl;

}

void show(){

cout<<"gadgets name"<<endl;

cout<<"launch attack"<<endl;

cout<<"doemon fly:"<<endl;

}

};

class benten:public game{

public:

int powers;

int chargewatch;

int watch;

void display(){

cout<<"watch name:"<<endl;

cin>>watch;

cout<<"list of name powers:"<<endl;

cin>>powers;

cout<<"total charge of the watch:"<<endl;

cin>>chargewatch;

}

void action(){

cout<<"rotate watch:"<<endl;

cout<<"benten fight scene:"<<endl;

cout<<"drive:"<<endl;

}

void show(){

cout<<"they can jump"<<endl;

cout<<"they can eat:"<<endl;

cout<<"they can walk:"<<endl;

}

};

int main(){

doremon d1;

d1.display();

d1.action();

d1.show();

benten d2;

d2.display();

d2.action();

d2.show();

}