Institute of Computer Technology

B. Tech Computer Science and Engineering

Subject: ESFP-II (2CSE203)

**PRACTICAL-7**

**AIM: - To learn about Inheritance in C++.**

**1. Implement a class government having two subclasses finance and defense. Now make a child Minister which should have finance and defense as parent class. Classes Government must possess a function Budget which display the message you will get budget less than 4000 crores. Make the object of Minister and access the function budget.**

***CODE:***

#include <iostream>

using namespace std;

class Government{

public:

void Budget()

{

cout<<"\nYou will get budget less than 4000 crores.";

}

};

class Finance : public Government {

};

class Defence : public Government {

};

class Minister : public Finance,Defence{

};

int main()

{

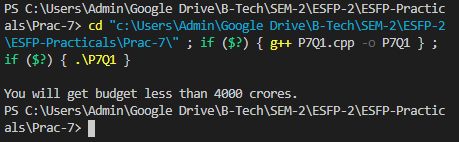
Minister obj;

obj.Finance::Budget();

return 0;

}

***OUTPUT:***

****

**2. Create two classes, A and B, with default constructors that announce themselves. Inherit a new class called C from A, and create a member object of B in C, but do not create a constructor for C. Create an object of class C and observe the results.**

***CODE:***

#include<iostream>

using namespace std;

class A

{

public:

A()

{

cout<<"\nThis is class A";

}

};

class B

{

public:

B()

{

cout<<"\nThis is class B.";

}

};

class C:public A

{

public:

B obj;

};

int main()

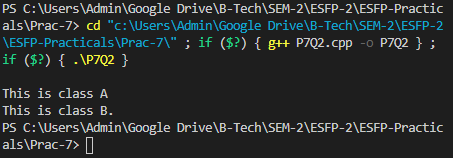
{

C obj1;

return 0;

}

***OUTPUT:***

******

**3. Implement a following scenario using C++ Inheritance: (Use Name & Eye color Data member to print the results)**

****

***CODE:***

#include <iostream>

using namespace std;

class MyFather{

public:

void F(){

cout<<"\nFather Name: Prakash";

cout<<"\nEye Color: Green";

}

};

class MyMother{

public:

void M(){

cout<<"\nMother Name: Sangita";

cout<<"\nEye Color: Dark-Brown";

}

};

class MySelf:public MyFather{

public:

void Y(){

cout<<"\nMy Name: Yash";

cout<<"\nEye Color: Dark-Brown";

}

};

class MySister:public MyMother,MyFather{

public:

void X(){

cout<<"\nSister Name: XYZ";

cout<<"\nEye Color: Black";

}

};

int main(){

MySelf obj;

obj.F();

obj.Y();

MySister obj1;

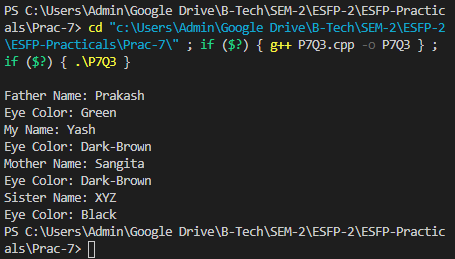
obj1.M();

obj1.X();

return 0;

}

***OUTPUT:***

****

**Post Practical Work**

**1. Write a Program in C++ to illustrate the order of execution of constructors and destructors in inheritance.**

***CODE:***

#include<iostream>

using namespace std;

class A{

public :

A()

{

cout<<"First Constructor"<<endl;

}

~A()

{

cout<<"\nFirst Destructor"<<endl;

}

};

class B : public A{

public :

B()

{

cout<<"\nSecond Constructor"<<endl;

}

~B()

{

cout<<"\nSecond Destructor"<<endl;

}

};

class C : public B{

public :

C()

{

cout<<"\nThird Constructor"<<endl;

}

~C()

{

cout<<"\nThird Destructor"<<endl;

}

};

int main()

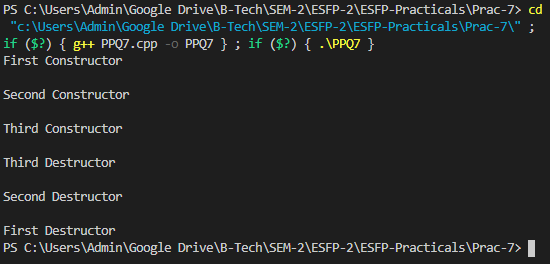
{

C obj;

return 0;

}

***OUTPUT:***

****

**2. #include <iostream>;**

**using namespace std;**

**class Info**

**{**

**char\* name;**

**int Number;**

**public:**

**void getInfo()**

**{**

**cout << "Info::getInfo";**

**getName();**

**}**

**void getName()**

**{**

**cout << "Info::getName";**

**}**

**};**

**class Name: public Info**

**{**

**char \*name;**

**public:**

**void getName()**

**{**

**cout << "Name::getName";**

**}**

**};**

**void main()**

**{**

**Info \*P;**

**Name n;**

**P = n;**

**p->getInfo();**

**}**

***ERRORS:***

* Termination in include.
* As P is pointer it requires address.
* Syntax error in last line.

***CORRECTED CODE:***

#include <iostream>

using namespace std;

class Info

{

char\* name;

int Number;

public:

void getInfo()

{

cout << "Info::getInfo";

getName();

}

void getName()

{

cout << "Info::getName";

}

};

class Name: public Info

{

char \*name;

public:

void getName()

{

cout << "Name::getName";

}

};

int main()

{

Info \*P;

Name n;

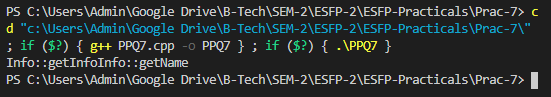
P = &n;

P->getInfo();

return 0;

}

***OUTPUT:***

****

**3.**

**Find the output:**

**#include<iostream>**

**using namespace std;**

**class base {**

**int arr[10];**

**};**

**class b1: public base { };**

**class b2: public base { };**

**class derived: public b1, public b2 {};**

**int main(void)**

**{**

**cout << sizeof(derived);**

**return 0;**

**}**

A.40

**B.80**

C. 0

D. 4

***OUTPUT:***

