**CODE:-**

# include <iostream>

using namespace std;

class Base

{

public:

virtual void display()

{

cout<<"Display base class."<<endl;

}

virtual void show()

{

cout<<"Show base class."<<endl;

}

};

class Derived:public Base

{

public:

void display()

{

cout<<"Display derived class."<<endl;

}

void show()

{

cout<<"Show derived class."<<endl;

}

};

int main()

{

Base B;

Derived D;

Base \*bptr;

bptr=&B;

cout<<"bptr points to base."<<endl;

bptr->display();

bptr->show();

bptr=&D;

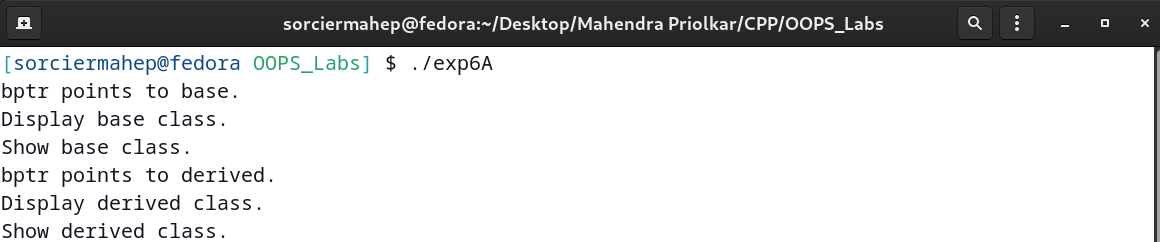
cout<<"bptr points to derived."<<endl;

bptr->display();

bptr->show();

return 0;

}

**OUTPUT:-**

**CODE:-**

# include <iostream>

using namespace std;

class media

{

public:

virtual void display()=0; //Pure virtual function

};

class book: public media

{

public:

void display()

{

cout<<"Book name is the Silmarillion."<<endl;

}

};

class tape: public media

{

public:

void display()

{

cout<<"Tape name is Star Wars Return Of The Jedi."<<endl;

}

};

int main()

{

media \* bptr[2];

book D1;

tape D2;

bptr[0]=&D1;

bptr[1]=&D2;

bptr[0]->display();

bptr[1]->display();

return 0;

}

**OUTPUT:-**

