G. H. Raisoni College Of Engineering And Management, Wagholi Pune

2021- 2022

Assignment no :- 7

Department	CE [SUMMER 2022 (Online)]		
Term / Section	III/B	Date Of **submission**	15-11-2021
Subject Name /Code	Object Oriented Programming/ UTIL201/UITP201		
Roll No.	SCOB77	Name	Pratham Rajkumar pitty
Registration Number	2020AC0E1100107		

Assignment no 7.

Aim :→ A book shop sells both books and video tapes. creat a class media that stores the title and price of the publication create two derived classes, one for storing number of pages in the book and another for storing playing time of tape. A function display() must be defined in all classes to display class contents. write a program using polymorphism and virtual function.

Theory :→

- Polymorphism :→ means having many forms

Polymorphism in C++ is defined as the capability of objects of different types responding to functions with same name in their own ways

It lets us create function with same name but different arguments, which will perform different actions.
i.e. functions with same name, but functioning in different way.

C++ supports operator overloading and function overloading.

Polymorphism is extensively used in implementing inheritance

Virtual Functions

- A virtual Function is a member Function which within a base class and is re-defined (overridden) by a derived class.

- when we refer to a derived class object using a pointer or a reference to the base class, we can call a virtual Function for that object and execute the derived class's version of the functions.

- Virtual Function ensure that the correct Function is called for an object, regardless of the type of reference (or pointer) used for function call.

- They are mainly used to achieve Runtime polymorphism.

- Functions are declared with a virtual keyword in base class.

- The resolving of function call is done at Run-time

Rules For Virtual Functions

1. Virtual Functions cannot be static.
2. A virtual Function cannot be a friend Function of another class.
3. Virtual Functions should be accessed using pointer or reference of base class type to

achieve runtime polymorphism.

4. The prototype of virtual functions should be the same in the base as well as derived class.
5. They are always defined in the base class and overridden in a derived class. it is not mandatory for the derived class to override (or re-define the virtual function), in that case, the base class version of the function is used.
6. A class may have virtual destructor but it cannot have a virtual constructor.

Ex. C++ virtual funtion simple ex.

```
# include <iostream>
using name space std;
class base {
 public:
  virtual void print()
  { cout << " print base class" << endl; }
  virtu void show()
  { cout << "show base class" << endl; } };
class derived : public base {
 public:
        void print()
        { cout << " print derived class" << endl; }
        void show()
        { cout << " show derived class" << endl; } };
int main()
{ base* bptr;
  derived d;
  bptr = &d;
```

// virtual Function, binded at runtime
bptr → print();

// non-virtual function, binded at compile time
bptr -> show();

Output:

print derived class
Show base class.

Explanation :→ In above code, base class pointer 'bptr' contains the address of object 'd' of derived class.

Late binding (Runtime) is done in accordance with the content of pointer (i.e location pointer to pointer) and Early binding [Compile time] is done in accordance ~~with~~ ~~to~~ to the type of pointer.

Since print() function is declared with virtual keyword so it will be bound at run-time (output is printed class as pointer is pointing to pointing to object of derived class and

Show() is non-virtual so it will be bound during compile time (output is shown base as pointer is of base type)

Program code:-

```
#include <iostream>
using namespace std;
class publication
{
   char title[10];
   float price;

public:
  virtual void read()
   {
      cout << "\nEnter Title of the Publication :";
      cin >> title;
      cout << "\nEnter the price of Publication :";
      cin >> price;
   }
   virtual void display()
   {
      cout << "Title is :" << title << "\n";
      cout << "Price is :" << price << "\n";
   }
};
class Storing_book_pages : public publication
{
   int page_count;

public:
   void read()
   {
```

```
        cout << "\nEnter the page count of book :";
        cin >> page_count;
    }
    void display()
    {
        cout << "Page count of book is :" << page_count << endl;
    }
};
class Storing_VideoTape_Time : public publication
{
    float playing_time;

public:
    void read()
    {
        cout << "\nEnter playing time in minutes :";
        cin >> playing_time;
    }
    void display()
    {
        cout << "Playing time in minutes :" << playing_time << endl;
    }
};
int main()
{
    cout << "\n\n--------Welcome to pratham's book shop---------" << endl;
    cout << "\nSCOB77_Pratham_Pitty_OOP_Assignment7\n"
         << endl;

    publication *ptr;
    publication p;
```

```
    p.read();
    p.display();

    cout << "-------------------------------------";

    Storing_book_pages b;
    ptr = &b;
    ptr->read();
    ptr->display();

    cout << "-------------------------------------";

    Storing_VideoTape_Time t;
    ptr = &t;
    ptr->read();
    ptr->display();
    return 0;
}
```

Output :-

```
"C:\Users\prath\OneDrive\Desktop\code block 1\SCOB77_Pratham_Pitty_OOP_Assignment7\bin\Debug\SCOB77_Pratham_Pitty_OOP_...

--------Welcome to pratham's book shop---------

SCOB77_Pratham_Pitty_OOP_Assignment7


Enter Title of the Publication :pratham

Enter the price of Publication :1001
Title is :pratham
Price is :1001
-------------------------------------
Enter the page count of book :50
Page count of book is :50
-------------------------------------
Enter playing time in minutes :60
Playing time in minutes :60

Process returned 0 (0x0)   execution time : 14.538 s
Press any key to continue.
```