RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF ENGINEERING & TECHNOLOGY

CSPS103: Object Oriented Programming

Lecture-20

Preeti Singh
Department of Computer Science & Engineering
Rama University, Kanpur

preeti.ru@ramauniversity.ac.in

OBJECTIVES

In this lecture, you will learn to:

- ❖Single Inheritance
- ❖Program
- ❖Ambiguity in single Inheritance
- ❖Single Level Inheritance Example: Inheriting Fields

SINGLE INHERITANCE

- The process in which a derived class inherits traits from only one base class, is called single inheritance.
- In single inheritance, there is only one base class and one derived class.
- The derived class inherits the behavior and attributes of the base class.
- The derived class can add its own properties i.e. data members (variables) and functions.
- It can extend or use properties of the base class without any modification to the base class.
- We declare the base class and derived class as given below:

```
class base_class
{
};
class derived_ class : visibility-mode base_ class
{
};
```

Program to illustrate concept of single inheritance

```
#include<iostream.h>
#include<conio.h>
class base //base class
{
int x,y;
public:
 void show() {
 cout<<"In base class";
 }
};
class derived : public base //derived class
{
int a,b;
public:
 void show2() {
 cout<<"\nIn derived class";
 }
};
int main() {
 derived d;
 d.show(); //uses base class's show() function
 d.show2(); //uses derived class's show2() function
 getch();
 return 0;
}
```

AMBIGUITY IN SINGLE INHERITANCE

- Whenever a data member and member functions are defined with the same name in both the base and derived class, ambiguity occurs.

- The scope resolution operator must be used to refer to particular class as:

object name.class name :: class member

SINGLE LEVEL INHERITANCE EXAMPLE: INHERITING FIELDS

```
#include <iostream.h>
 class Account {
   public:
   float salary = 60000;
 };
   class Programmer: public Account {
   public:
   float bonus = 5000;
   };
int main(void) {
     Programmer p1;
     cout<<"Salary: "<<p1.salary<<endl;
     cout<<"Bonus: "<<p1.bonus<<endl;
    return 0;
}
```

REFERENCES

- Kernighan, Brian W., and Dennis M. Richie. The C Programming Language. Vol. 2. Englewood Cliffs: Prentice-Hall, 1988.
- King, Kim N., and Kim King. C programming: A Modern Approach. Norton, 1996.
- Bjrane Stroustrup, “C++ Programming language”,3rd edition, Pearson education Asia(1997)
- Lafore R.”Object oriented Programming in C++”,4th Ed. Techmedia,New Delhi(2002).
- Yashwant Kenetkar,”Let us C++”,1stEd.,Oxford University Press(2006)
- B.A. Forouzan and R.F. Gilberg,CompilerScience,”A structured approach using C++” Cengage Learning, New Delhi.
- https://www.javatpoint.com/cpp-tutorial
- https://www.tutorialspoint.com/cplusplus/index.htm
- https://ambedkarcollegevasai.com/wp-content/uploads/2019/03/CPP.pdf
- https://onlinecourses.nptel.ac.in/noc20_cs07/unit?unit=3&lesson=19

Multiple Choice Question:

Q1. Members which are not intended to be inherited are declared as _____

a) Public members

b) Protected members

c) Private members

d) Private or Protected members

Multiple Choice Question:

Q2. While inheriting a class, if no access mode is specified, then which among the following is true? (in C++)

a) It gets inherited publicly by default

b) It gets inherited protected by default

c) It gets inherited privately by default

d) It is not possible

Multiple Choice Question:

Q3. If a derived class object is created, which constructor is called first?

a) Base class constructor

b) Derived class constructor

c) Depends on how we call the object

d) Not possible

Multiple Choice Question:

Q4. How can you make the private members inheritable?

a) By making their visibility mode as public only

b) By making their visibility mode as protected only

c) By making their visibility mode as private in derived class

d) It can be done both by making the visibility mode public or protected

Multiple Choice Question:

Q5. How many types of inheritance are possible in C++?

a) 2

b) 3

c) 4

d) 5

Summary

In this lecture, you learned that:

- When one class inherits another class, it is known as single level inheritance