



RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF ENGINEERING & TECHNOLOGY

CSPS103: Object Oriented Programming

Lecture-12

Preeti Singh

Department of Computer Science & Engineering
Rama University, Kanpur

preeti.ru@ramauniversity.ac.in

OBJECTIVES

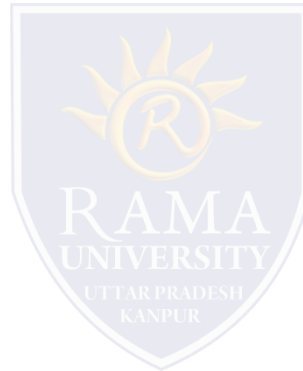
In this lecture, you will learn to:

- ❖ **Scope Resolution operator**

- ❖ **Static**

- ❖ **Static Field**

- ❖ **Static Member Functions**



SCOPE RESOLUTION OPERATOR

- ❑ Member functions can be defined within the class definition or separately using scope resolution operator (::).
- ❑ Defining a member function within the class definition declares the function inline, even if you do not use the inline specifier.
- ❑ Defining a member function using scope resolution operator uses following declaration

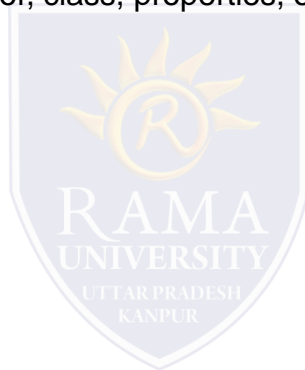
```
return-type class-name::func-name(parameter- list)  
{  
// body of function  
}
```

```
Class myclass {  
int a;  
public:  
void set_a(intnum); //member function declaration  
int get_a( ); //member function declaration  
};  
//member function definition outside class using scope resolution operator  
void myclass :: set_a(intnum)  
{  
a=num;  
}  
int myclass::get_a( ) {  
return a;  
}
```



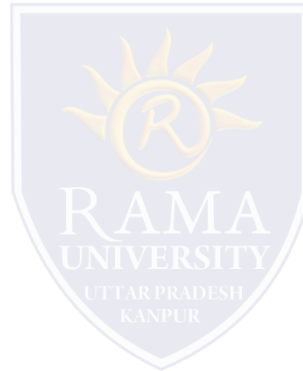
What is static

- ❑ In C++, static is a keyword or modifier that belongs to the type not instance.
- ❑ So instance is not required to access the static members.
- ❑ In C++, static can be field, method, constructor, class, properties, operator and event.



STATIC FIELD

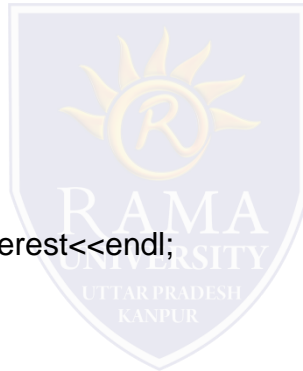
- ☐ A field which is declared as static is called static field.
- ☐ Unlike instance field which gets memory each time whenever you create object, there is only one copy of static field created in the memory.
- ☐ It is shared to all the objects.



PROGRAM : STATIC FIELD

```
#include <iostream.h>
class Account {
public:
    int accno; //data member (also instance variable)
    string name; //data member(also instance variable)
    static float rateOfInterest;
    Account(int accno, string name)
    {
        this->accno = accno;
        this->name = name;
    }
    void display()
    {
        cout<<accno<< "<<name<< " "<<rateOfInterest<<endl;
    }
};

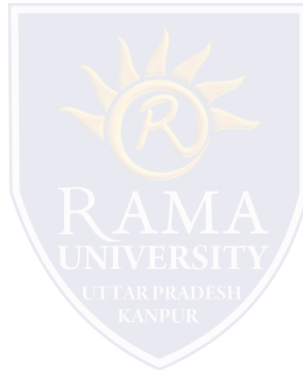
float Account::rateOfInterest=6.5;
int main(void) {
    Account a1 =Account(201, "Sanjay"); //creating an object of Employee
    Account a2=Account(202, "Nakul"); //creating an object of Employee
    a1.display();
    a2.display();
    return 0;
}
```



STATIC MEMBER FUNCTIONS

- ❑ Member functions may also be declared as static.
- ❑ They may only directly refer to other static members of the class.
- ❑ A static member function can be called using the class name instead of its objects as follows:

class name :: function name



PROGRAM : STATIC CLASS MEMBERS

//Program showing working of static class members

```
#include <iostream.h>
#include<conio.h>
class static_type {
static int i; //static data member
public:
static void init(int x) {i = x;} //static member function
void show() {cout << i;};
int static_type :: i; // static data member definition
int main(){
static_type::init(100); //Accessing static function
static_type x;
x.show();
return 0;
}
```



REFERENCES

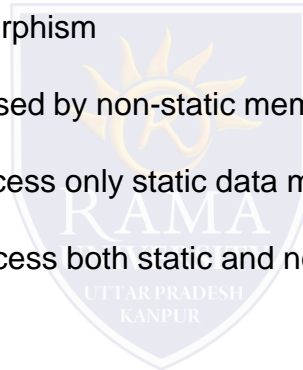
- Kernighan, Brian W., and Dennis M. Richie. The C Programming Language. Vol. 2. Englewood Cliffs: Prentice-Hall, 1988.
- King, Kim N., and Kim King. C programming: A Modern Approach. Norton, 1996.
- Bjarne Stroustrup, "C++ Programming language", 3rd edition, Pearson education Asia (1997)
- Lafore R.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

MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q1. Which of the following is correct about static variables?

- a) Static functions do not support polymorphism
- b) Static data members cannot be accessed by non-static member functions
- c) Static data members functions can access only static data members
- d) Static data members functions can access both static and non-static data members



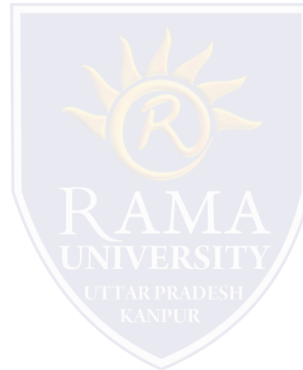
MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q2. Const qualifier can be applied to which of the following?

- i) Functions inside a class**
- ii) Arguments of a function**
- iii) Static data members**
- iv) Reference variables**

- a) i, ii and iii
- b) i, ii, iii, and iv
- c) ii, iii and iv
- d) i only

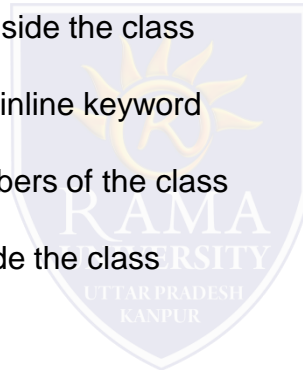


MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q3. Which functions of a class are called inline functions?

- a) All the functions containing declared inside the class
- b) All functions defined inside or with the inline keyword
- c) All the functions accessing static members of the class
- d) All the functions that are defined outside the class

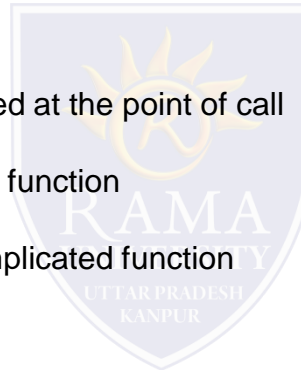


MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q4. Pick the incorrect statement about inline functions in C++?

- a) They reduce function call overheads
- b) These functions are inserted/substituted at the point of call
- c) Saves overhead of a return call from a function
- d) They are generally very large and complicated function

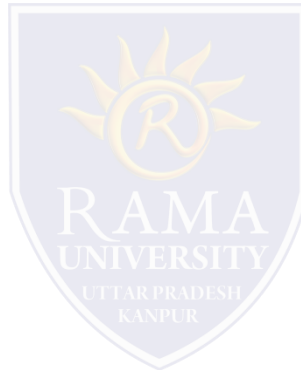


MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q5. Inline functions are avoided when _____

- a) function contains static variables
- b) function have recursive calls
- c) function have loops
- d) all of the mentioned



Summary

In this lecture, you learned that:

- Concept of scope resolution operator
- Concept of static
 - 1) Static Field
 - 2) Static Member Functions

