



FACULTY OF ENGINEERING & TECHNOLOGY

CSPS103: Object Oriented Programming

Lecture-14

Preeti Singh

Department of Computer Science & Engineering
Rama University, Kanpur

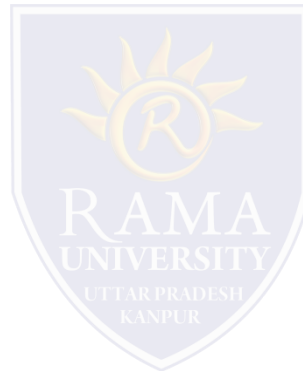
preeti.ru@ramauniversity.ac.in

OBJECTIVES

In this lecture, you will learn to:

❖ **Copy Constructors**

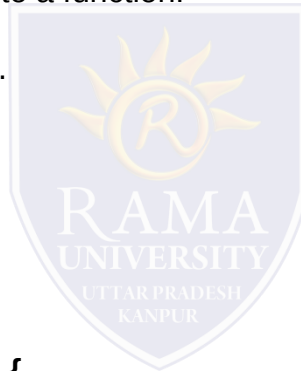
❖ **Destructor**



COPY CONSTRUCTORS

The copy constructor is a constructor which creates an object by initializing it with an object of the same class, which has been created previously. If class definition does not explicitly include copy constructor, then the system automatically creates one by default. The copy constructor is used to:

- ☐ Initialize one object from another of the same type.
- ☐ Copy an object to pass it as an argument to a function.
- ☐ Copy an object to return it from a function.

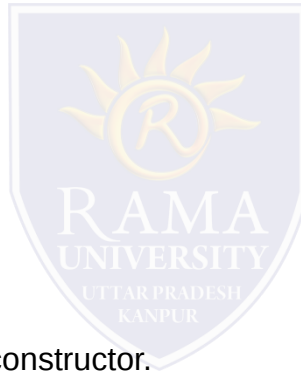


Syntax

```
classname (const classname &obj) {  
  
    // body of constructor  
  
}
```

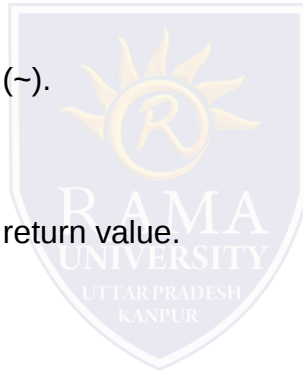
PROGRAM: COPY CONSTRUCTORS

```
#include <iostream.h>
class A
{
public:
    int x;
    A(int a)           // parameterized constructor.
    {
        x=a;
    }
    A(A &i)             // copy constructor
    {
        x = i.x;
    }
};
int main()
{
    A a1(20);          // Calling the parameterized constructor.
    A a2(a1);           // Calling the copy constructor.
    cout<<a2.x;
    return 0;
}
```



DESTRUCTOR

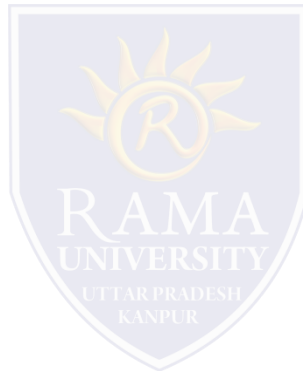
- ☐ A destructor destroys an object after it is no longer in use.
- ☐ The destructor, like constructor, is a member function with the same name as the class name.
- ☐ But it will be preceded by the character Tilde (~).
- ☐ A destructor takes no arguments and has no return value.
- ☐ Each class has exactly one destructor.
- ☐ If class definition does not explicitly include destructor, then the system automatically creates one by default.
- ☐ It will be invoked implicitly by the compiler upon exit from the program to clean up storage that is no longer accessible.



PROGRAM: DESTRUCTOR

// A Program showing working of constructor and destructor

```
#include<iostream.h>
#include<conio.h>
class Myclass
{
public:
int x;
Myclass()
{ //Constructor
x=10;
}
~Myclass()
{ //Destructor
cout<<"Destructing...." ;
}
int main()
{
Myclass ob1, ob2;
cout<<ob1.x<<" "<<ob2.x;
return 0;
}
```



REFERENCES

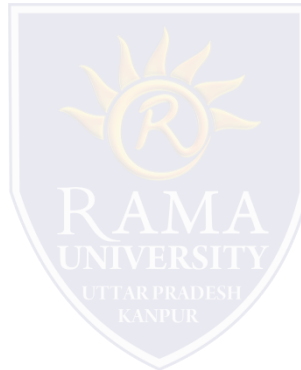
- Kernighan, Brian W., and Dennis M. Richie. The C Programming Language. Vol. 2. Englewood Cliffs: Prentice-Hall, 1988.
- King, Kim N., and Kim King. C programming: A Modern Approach. Norton, 1996.
- Bjarne Stroustrup, "C++ Programming language", 3rd edition, Pearson education Asia (1997)
- Lafore R. "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://ambedkarcollegevasai.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. How many types of constructors are there in C++?

- a) 1
- b) 2
- c) 3
- d) 4

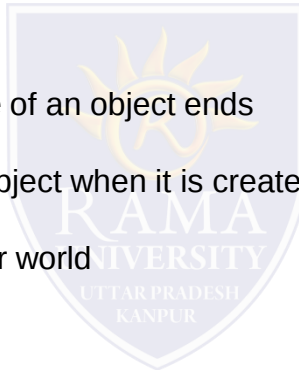


MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q2. What is the role of destructors in Classes?

- a) To modify the data whenever required
- b) To destroy an object when the lifetime of an object ends
- c) To initialize the data members of an object when it is created
- d) To call private functions from the outer world



MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q3. What is syntax of defining a destructor of class A?

- a) `A(){}`
- b) `~A(){}`
- c) `A::A(){}`
- d) `~A(){};`

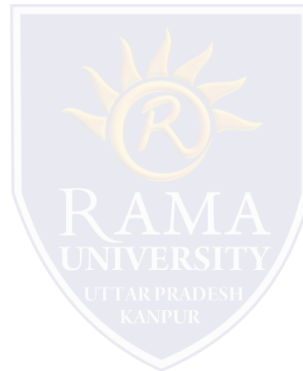


MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q4. When destructors are called?

- a) When a program ends
- b) When a function ends
- c) When a delete operator is used
- d) All of the mentioned

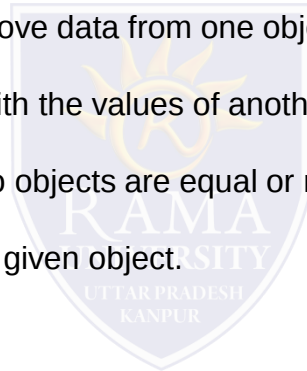


MULTIPLE CHOICE QUESTION

Multiple Choice Question:

Q5. What is a copy constructor?

- a) A constructor that allows a user to move data from one object to another
- b) A constructor to initialize an object with the values of another object
- c) A constructor to check the whether to objects are equal or not
- d) A constructor to kill other copies of a given object.



Summary

In this lecture, you learned that:

- A Copy constructor is an overloaded constructor used to declare and initialize an object from another object
- A destructor works opposite to constructor.

