**Global Group of Institutions**

**Demo Question Paper – Set – IV**

**Subject – Object Oriented Programming with C++**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Questions Lists - Constructor And Destructor** | **Options** |
| 1. | Which of the followings is/are automatically added to every class, if we do not write our own.  A. Copy Constructor.  B. Assignment Operator  C. A constructor without any parameter  D. All of the above | D |
| 2. | Which of the following gets called when an object is being created?  A. Constructor B. Virtual Function  C. Destructors D. Main | A |
| 3. | Destructor has a same name as the constructor and it is preceded by?  A. ! B. ? C. ~ D. $ | C |
| 4. | Like constructors, can there be more than one destructors in a class?  A. Yes B. No C. May Be D. Can't Say | B |
| 5. | State whether the following statements about the constructor are True or False.  **I) constructors should be declared in the private section.**  **II) constructors are invoked automatically when the objects are created.**  A. True, True B. True, False  C. False, True D. False, False | C |
| 6. | Which of the following is true about constructors?  **i) They cannot be virtual**  **ii) They cannot be private.**  **iii) They are automatically called by new operator.**  A. All i,ii,iii B. i & iii C. ii & iii D. i & ii | B |
| 7. | Destructors \_\_\_\_\_\_\_\_\_\_ for automatic objects if the program terminates with a call to function exit or function abort  A. Are called B. Are not called  C. Are inherited D. Are created | B |
| 8. | Which constructor function is designed to copy object of same class type?  A. Copy constructor B. Create constructor  C. Object constructor D. Dynamic constructor | A |

|  |  |  |
| --- | --- | --- |
| 9. | What will be the output of the following program?  **#include<iostream.h>**  **using namespace std;**  **class LFC**  **{**  **int id;**  **static int count;**  **public:**  **LFC() {**  **count++;**  **id = count;**  **cout << "constructor for id " << id << endl;**  **}**  **~LFC() {**  **cout << "destructor for id " << id << endl;**  **}**  **};**  **int LFC::count = 0;**  **void main()**  **{**  **LFC a[3];**  **}**  A. constructor for id 1 constructor for id 2 constructor for id 3 destructor for id 3 destructor for id 2 destructor for id 1  B. constructor for id 1 constructor for id 2 constructor for id 3 destructor for id 1 destructor for id 2 destructor for id 3  C. Compiler Dependent  D. constructor for id 1 destructor for id 1 | D |
| 10. | What will be the output of the following program?  **#include <iostream>**  **using namespace std;**  **class LFC {**  **LFC() { cout << "Constructor called"; }**  **};**  **void main() {**  **LFC t1;**  **}**  A. Compiler Error B. Runtime Error  C. Constructor called D. destructor for id 1 | A |

======================================================