**Global Group of Institutions**

**Demo Question Paper – Set – XIV**

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

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Questions Lists – Friend Function** | **Options** |
| 1. | A friend class can access \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ members of other class in which it is declared as friend.  A. private  B. protected  C. public  D. Both A and B | D |
| 2. | A friend function can be  A. A method of another class  B. A global function  C. Both A and B  D. None of the above | C |
| 3. | If class A is a friend of B, then B doesn’t become a friend of A automatically.  A. TRUE  B. FALSE  C. Can be true and false  D. Cannot say | A |
| 4. | Which of the following is false?  A. Friendship is not inherited  B. The concept of friends is there in Java.  C. Both A and B  D. None of the above | B |
| 5. | Which keyword is used to represent a friend function?  A. friend B. Friend  C. friend\_func D. Friend\_func | A |
| 6. | Which of the following is correct about friend functions?  A. Friend functions use the dot operator to access members of a class using class objects  B. Friend functions can be private or public  C. Friend cannot access the members of the class directly  D. All of the above | D |
| 7. | Pick the correct statement.  A. Friend functions are in the scope of a class  B. Friend functions can be called using class objects  C. Friend functions can be invoked as a normal function  D. Friend functions can access only protected members not the private members | C |

|  |  |  |
| --- | --- | --- |
| 8. | Where does keyword ‘friend’ should be placed?  A. function declaration  B. function definition  C. main function  D. block function | A |
| 9. | What will be output for the following code?  **class Box**  **{**  **int capacity;**  **public:**  **void print();**  **friend void show();**  **bool compare();**  **friend bool lost();**  **};**  A. 1  B. 2  C. 3  D. 4 | B |
| 10. | What will be output for the following code?  **#include <iostream>**  **class A {**  **private:**  **int a;**  **public:**  **A() { a = 0; }**  **friend class B; // Friend Class**  **};**    **class B {**  **private:**  **int b;**  **public:**  **void showA(A& x) {**  **std::cout << ""A::a="" << x.a;**  **}**  **};**  **void main() {**  **A a;**  **B b;**  **b.showA(a);**  **}**  A. A::a=0  B. A  C. a=0  D. A::0 | A |

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