**Answer1:**

|  |  |
| --- | --- |
| Line 1: | Aclass() default |
| Line 2: | - |
| Line 3: | Aclass(int) |
| Line 4: | Bclass() default |
| Line 5: | Aclass(const &) i:5  func1  ~Aclass() i:5 |
| Line 6: | func2  Aclass(int)  ~Aclass() i:7 |
| Line 7: | Bclass(const &) |
| Line 8: | ~Aclass() i:5 |
| Line 9: | ~Bclass()  ~Bclass()  ~Aclass() i:0 |

**Answer2:**

1. next is a private member of myNode class and to be able to use it directly, myLinkedList class needs to be defined as a friend class of myNode. To make the given lines of code to work, “friend class myLinkedList;” must be added after line 4.

|  |
| --- |
| class myNode{  int data;  myNode \*next;  int id;  static int idCounter;  public:  myNode(int value) : data(value){  next = NULL;  idCounter++;  id=idCounter;  }  static void init\_counter(){ counter=0; } // to initialize idCounter  };   * “myNode::init\_counter();” line must be called to initialize idCounter when first node is created. |

1. 4 creation id=4

3 creation id=3

2 creation id=2

1 creation id=1

1. **The output:**

4 creation id=4

3 creation id=3

2 creation id=2

1 creation id=1

**The reason:** The copy constructor of myLinkedList class is not defined, so default copy constructor provided by the compiler is used when mylist2 is created as a copy of mylist. Default copy constructor simply equalizes data members without allocating necessary space for the data. Thus, the program gives runtime error at the end of the program when both destructors of mylist and mylist2 try to free the same memory space.

1. The copy constructor of myLinkedList must be defined as below:

|  |
| --- |
| myLinkedList(const myLinkedList& listToCopy){  head = NULL;  myNode \*ptr = listToCopy.head;  while(ptr){  add(ptr->data);  ptr = ptr->next;  }  } |

**Output of mylist2.print():**

1 creation id=8

2 creation id=7

3 creation id=6

4 creation id=5

**If add\_totheend() method is used instead of add(), the output of mylist2.print() will be:**

4 creation id=5

3 creation id=6

2 creation id=7

1 creation id=8