Session 5 Assignment1

Task 1. Write a simple Timer that can periodically print a timeout message.

Solution: Below is the Java program (also attaching JAVA file "ReminderBeep.java")

```java
import java.awt.Toolkit;

import java.util.Timer;

import java.util.TimerTask;


public class ReminderBeep {

 Toolkit toolkit;

 Timer timer;


 public ReminderBeep(int seconds) {

  toolkit = Toolkit.getDefaultToolkit();

  timer = new Timer();

  timer.schedule(new RemindTask(), seconds * 1000);

 }


 class RemindTask extends TimerTask {

  public void run() {

   System.out.println("Timed Out!!!");

   toolkit.beep();

   System.exit(0);

  }

 }
```
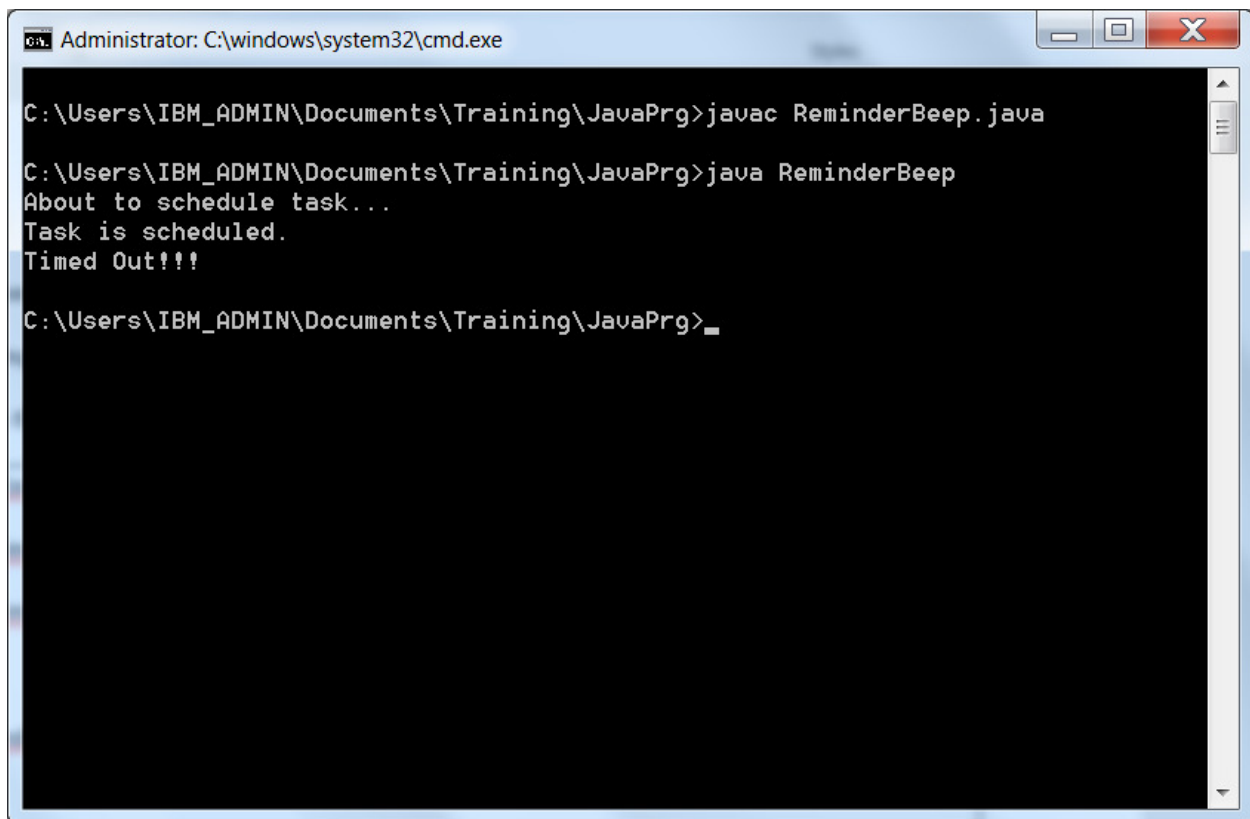
```
  public static void main(String args[]) {

    System.out.println("About to schedule task...");

    new ReminderBeep(5);

    System.out.println("Task is scheduled.");

  }

}
```

Output of the program:

Task 2. Write a program to build any collection containing duplicates. Create its copy with all duplicates removed.

Solution: Below is the Java program (also attaching JAVA program "LinkedList.java")

```java
class LinkedList {

  static Node head;

    static class Node {

       int data;

      Node next;

      Node(int d) {

        data = d;

        next = null;

      }

  }

   /* Function to remove duplicates from an unsorted linked list */

  void remove_duplicates() {

    Node ptr1 = null, ptr2 = null, dup = null;

    ptr1 = head;


    /* Pick elements one by one */

    while (ptr1 != null && ptr1.next != null) {

      ptr2 = ptr1;


      /* Compare the picked element with rest of the elements */

      while (ptr2.next != null) {
/* If duplicate then delete it */

        if (ptr1.data == ptr2.next.data) {
```

```java
            /* sequence of steps is important here */

            dup = ptr2.next;

            ptr2.next = ptr2.next.next;

            System.gc();

        } else {

            ptr2 = ptr2.next;

        }

    }

    ptr1 = ptr1.next;

  }

}


void printList(Node node) {

  while (node != null) {

    System.out.print(node.data + " ");

    node = node.next;

  }

}


public static void main(String[] args) {

  LinkedList list = new LinkedList();

  list.head = new Node(10);

  list.head.next = new Node(12);

  list.head.next.next = new Node(11);
```

```
    list.head.next.next.next = new Node(11);

    list.head.next.next.next.next = new Node(12);

    list.head.next.next.next.next.next = new Node(11);

    list.head.next.next.next.next.next.next = new Node(10);

  System.out.println("Linked List before removing duplicates : \n ");

    list.printList(head);

    list.remove_duplicates();

    System.out.println("");

    System.out.println("Linked List after removing duplicates : \n ");

    list.printList(head);

  }

}
```
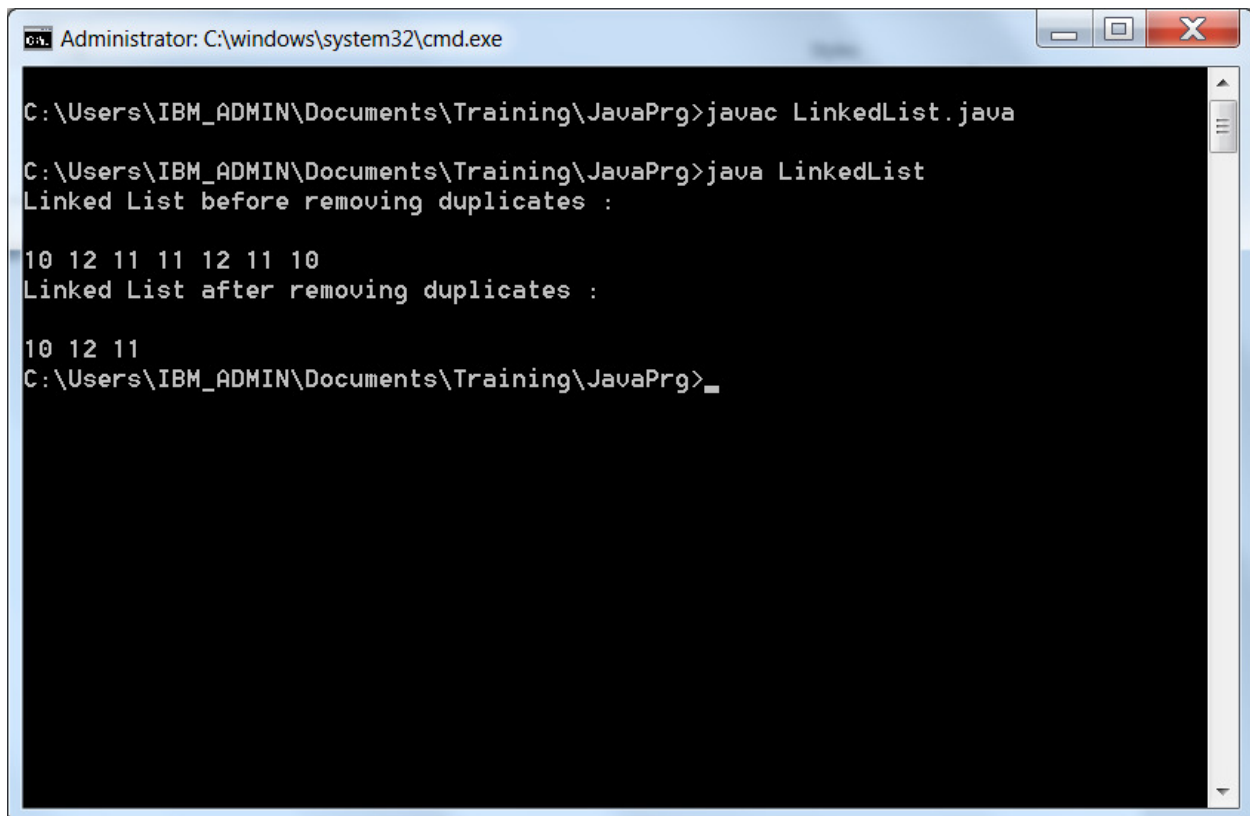
Output of the Program: