K. J. SOMAIYA INSTITUTE OF MANAGEMENT STUDIES AND RESEARCH

MCA –Semester II (Batch -2016-2019)

Subject: Core JAVA

(SEM-II End-Term Examination)

13/04/2017

Maximum Marks: 50
Duration: 2 hours

Note:
1. Q.1 is compulsory
2. Attempt any <u>four</u> from Q.2 to Q.7

Q.	a) Give the name of predefined final class.	(2)
1	b) What is meant by daemon thread? In java runtime, what is its role?	(2)
	c) Explain any 2 utility classes.	(2)
	d) Define prepared statements?	(2)
	e) Why constructors cannot be considered as member function?	(2)

Q.	a) How many types of JDBC drivers are present? Discuss them.	(04)
2	b) Discuss enum type with example	(04)
	c) Define deadlock and notify.	(02)

Q. Write a JDBC programs for the following using menu driven. (07)

3   a)Create an Address)-Book Table with the following

  fields: AddID(numeric), Name(Text), Email(Text) and PhoneNo.

  b) To insert n records in this table using prepared statements.

  c) To query this table -user enters the phone number and display the name and email-id.

  d) To draw a circle in center of the applet. Circle remains in center when applet (03) resized.

Q. a) Write a program of threads to show inter-communication of two threads: t1 and (06)

4   t2 synchronizing on a shared object. Let t1 print message "ping --->" and t2 print

  message "<--- pong". Make the threads execute in a way that the output displays a

consistent patterns of "ping ---> <--- pong".

b) Write a program which writes 10 numbers in the file. Read this file and identify (04) odd and even numbers among them.

Q.   **Write short notes on following.**

5   **a) Random Access File**   (03)

       b) Event delegation model.   (04)

       c) **Explain abstract, final and public class modifier.**   (03)

Q.   a) Explain the importance of Graphics, Color and Font class of java.awt package (04)

6   with its method.

       b)Explain getCodeBase and getDocumentBase methods with Example.   (04)

Q.   **a) AWT components are known as heavy weight component? Elaborate.**   (05)

7   **b) Explain the concept of creating anonymous class with appropriate example.** (05)

****