Birla Institute of Technology & Science – Pilani

First Semester 2016-17

Date: 13 November, 2016 (Sunday)

ONLINE TEST (OPEN BOOK)

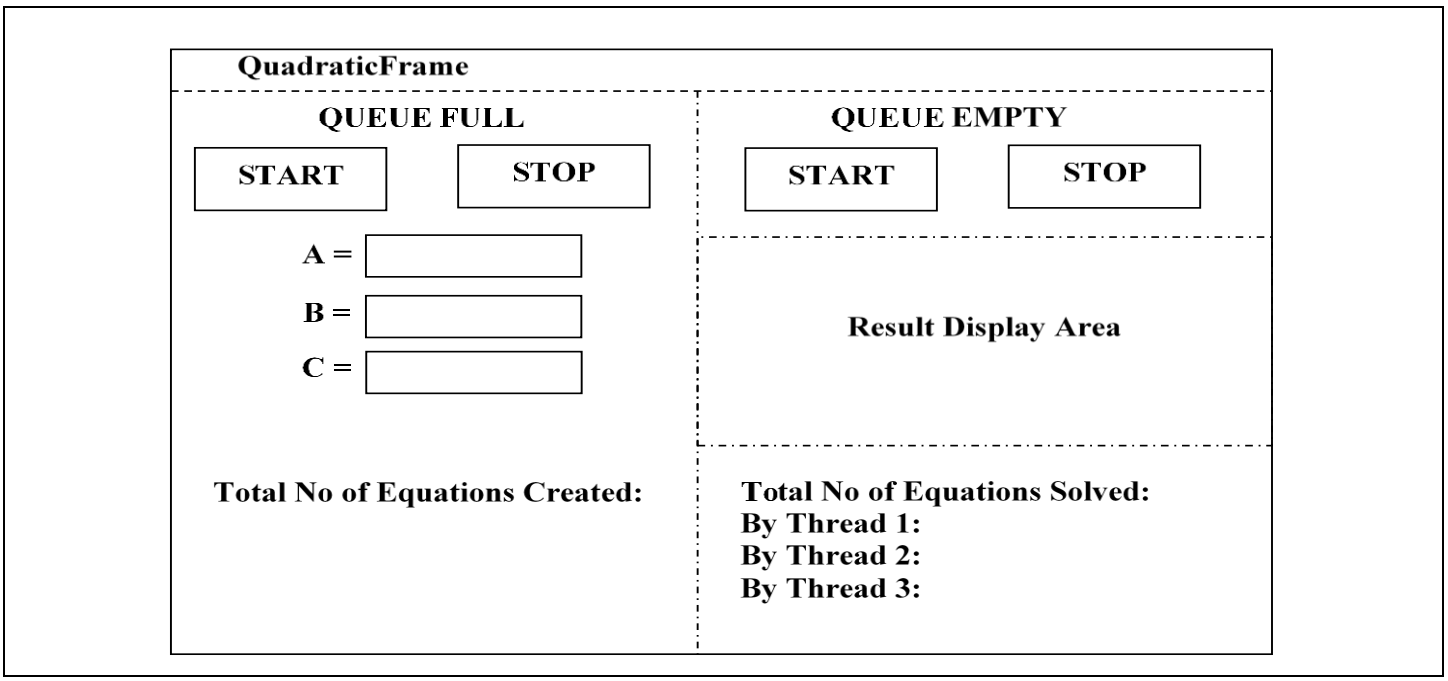
Course No and Name:	**Object Oriented Programming (CS F213)**		
Max. Marks: 90	**Weightage:30%**	**Time:**	**3:00 PM – 6:00 PM**

Important Instructions

- You are given a .java file named 'OnlineTest.Java'. Rename the file as P<IDNO>.java. For example if your ID NO is '2014A7005P' then the name of the renamed file should be 'P2014A7005.java'.
- The safekeeping of the written code is your responsibility. No query about the 'lost or misplaced' files will be considered at the end of the exam. Keep Saving your work regularly.

You are given the code for the class ***QuadraticEquation*** which encapsulates the general quadratic equation of the form $ax^2 + bx + c = 0$, where a, b & c are integer constants. [Assume the values for a, b & c is in range 0-20]. You are also given the code for the class ***QuadraticEquationQueue*** which encapsulates queue for QuaraticEquation objects with capacity 10. [Queue is implemented using arrays of fixed capacity 10]. All the necessary methods required for the implementation of queue are provided in the class.

Consider the following figure which shows a Frame with title "QuadraticFrame"



The overall frame area is divided into two main panels (left side and right side).

LeftSidePanel :

First Row of the left side contains a label (with text QUEUE FULL) which becomes green in color indicating that ***QuadraticEquationQueue*** is full**.** Next row shows two buttons with labels **START** and STOP. When **START**

button is pressed it starts one instance of thread named CreateThread and three text fields corresponding to labels A=,B= & C= starts displaying random integers in range 1-20 after 100 ms intervals apart. When START button is pressed for the first time then it starts the CreateThread and later on it only restarts the same thread only if it is stopped earlier otherwise pressing of the button is ignored. Similarly, when **STOP** button is pressed it temporarily stops the instance of CreateThread only if it is earlier started or resumed and all text fields stops displaying random integers otherwise pressing of the STOP button is ignored.

CreateThread does the following tasks whenever it executes:

(i) It reads the values displayed in the three text fields after every 150 ms interval, creates a QuadraticEquation instance and adds it into QuadraticEquationQueue instance. If any of the value(s) read from the three text fields is empty ("") OR the value from the text field corresponding to Label "A=" is 0 then no instance of QuadraticEquation is created and thread ignores all the values and continues with the next iteration.

(ii) If queue is already full then it waits for the queue to become a bit empty.

(iii) After addition if the queue becomes full, then it changes the color of QUEUE FULL label to green and when it is not full it changes the color of QUEUE FULL Label to black.

(iv) After every addition , it updates the bottom most label ("Total No of Equations Created") of left side Panel

RightSidePanel

First Row of the right side contains a label (with text QUEUE EMPTY) which becomes green in color when ***QuadraticEquationQueue*** is empty. When **START** button is pressed for the first time it starts three instance of thread named ***ComputeThreads*** each of which computes the roots of QuadraticEquation instance by removing it from the ***QuadraticEquationQueue*** instance. Successive pressing of the start button will only restarts the three ComputeThread instances only if they are stopped earlier otherwise pressing of the button is ignored. Similarly when STOP button is pressed it temporarily stops the ComputeThreads only if they are started or resumed earlier otherwise pressing of the STOP button is ignored. Result Display Area is composed of an array of 10 JLabel references and is used to display the result of the computed quadratic equation.

Each **ComputeThread** does the following tasks whenever it executes:

(i) It removes the QuadraticEquation reference from the QuadraticEquationQueue after every 150 ms inetrval and computes the roots of the equation.

(ii) Every ComputeThread maintains its own counter which indicates how many equations it has solved so far.

(iii) Whenever any of the ComputeThread(s) computes the roots of any equation , it updates

(a) The value of the Label "Total No of Equations Solved:"

(b) The value of its respective Labels in the bottom portion of the right side Panel.

(c) The value of the label (selected by obtaining the remainder by dividing the total no of equations created by 10) with result of the corresponding computed quadratic equation. [For More Details Look at the Demo]

(iv) If queue is empty then it waits for the queue to become a bit full.

(v) After removing element from QuadraticEquationQueue, if queue becomes empty then it changes the color of QUEUE EMPTY label to green.

Note:

1. To display random integers, you can write either a separate thread class or can use Timer class from java.
2. To generate random integers in range 1 – 20, use Random class form java.util as follows :
   Random random = new Random (20);
   int nextNo = random.nextInt(20); // generates random integer less than 20
3. You are given a sample executable java source file (Online.java) which on execution will display the frame with proper labels. This file contains the code for the classes "QuadraticEquation", "QuadraticEquationQueue" and partial OnlineMain (Driver class).

4. You can modify the code for the given classes if required.
5. One QuadraticEquation object should be computed only by one ComputeThread.

Tasks to be performed:

1. Implement the thread classes CreateThread, ComputeThread as per requirements given in the above specifications.
2. Modify the Driver class by writing suitable actionlistners for corresponding buttons for left side Panel and right side Panel. Modify it in such a way that it works as per specifications.

Important Instructions:

1. ***Save the given file as P<YourFullIdno>.java. Make sure You use capital letters for discipline while saving the file. E.g 2015A7PS134P.java, 2014B1A7675.java etc.***
2. ***All the code You have to write only in a single source java file.***
3. ***Keep on Saving your work regularly. You are responsible for the safekeeping of your work***