DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
B.TECH – SEMESTER – V, IT
SUBJECT: [IT 510] Core Java Technology

Examination	**:**	**Second Sessional**	**Seat No.**	**:**	
Date	**:**	**07/09/2018**	**Day**	**:**	**Friday**
Time	**:**	**11:45 to 01:00 PM**	**Max. Marks**	**:**	**36**

INSTRUCTIONS:
1. Figures to the right indicate maximum marks for that question.
2. The symbols used carry their usual meanings.
3. Assume suitable data, if required & mention them clearly.
4. Draw neat sketches wherever necessary.

Q.1 **Do as directed.** **[12]**

(a) Write names of event handlers of WindowListener class. [2]

(b) Create your own button class that allows us to create a button having red color with blue color string or caption. [2]

(c) Discuss uses of Panel and Canvas classes in GUI application. [2]

(d) Draw and explain Thread State Transition diagram. [2]

(e) Explain: checked exception and un-checked exception. [2]

(f) State True/False and also write the justification for your answer: [2]

1. "Finally Block does not execute always".
2. "Daemon thread is a low priority thread which runs in the background.

Q.2 Attempt ***Any Two*** from the following questions. **[12]**

(a) 1. Explain use of Font and FontMetrics class using suitable example. [2]

2. Using appropriate layout managers, prepare the layout as shown in the figure (Fig. Q.2a) for a GUI application. In each rectangle, you can place a button component with the string OK. [4]

(b) Using AWT, write a program that displays a circle, filled with red color, of radius 30 pixels in the center of the application. When the user presses any of the arrow keys (←, →, ↑, ↓), the circle moves in that direction by 10 pixels. When the user clicks using left mouse-button on left, right, up, or down side of the center of the circle, the circle moves in the clicked direction by 5 pixels. Using mouse, the user may not click exactly on a point falling on imaginary horizontal line or vertical line passing through the center of the circle. Refer the figure (Fig.Q.2b), horizontal and vertical lines are imaginary, not to be drawn by your application. For example, when the user clicks on point A or B, the circle should move in right direction, and if user clicks of C or D, the circle should move in up direction. Likewise, you can assume for left and down direction. [6]

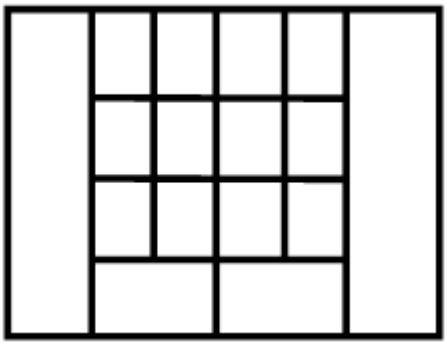

Fig. Q.2a

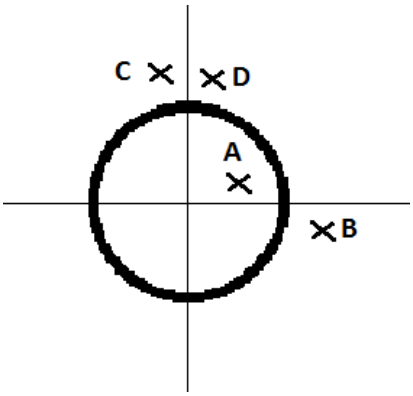

Fig. Q.2b

(c) 1. What is event driven programming and event delegation model in Java? [6]

2. Discuss concepts of event, event Source, and event listener using suitable program code.

3. Using suitable code explain use of adapter class for handling events of window.

Q.3 Attempt ***ALL*** from the following questions. **[12]**

(a) Discuss about how exceptions are handled in Java and also write a java program that raises an exception named InsufficientFundsException whenever a customer of a bank tries to withdraw an amount higher than the available amount. [5]

(b) Create an Application which moves a simple moving banner i.e. It should displays a string "DDUniversity, Nadiad" continuously, moving on left side of the frame until it touches to left side (i.e x=0), and reverses its direction moving on right side until it touches to right side of the frame(i.e x=max) . [5]

(c) Answer the following in the context of Exception handling: [2]
1. What is relation between Throwable, Error, and Exception
2. When we use multiple catch clauses for exception classes having inheritance, in what sequence the catch clauses should be written?

OR

Q.3 Attempt ***ALL*** from the following questions. **[12]**

(a) Write the output for the code snippet given below. [4]

1.
```
class A extends Thread {
     public void run(){
          System.out.println("Status:"+
                                   isAlive());
     }
}

public class Test{
    public static void main(String[] args) {
            A obj=new A();
            obj.start();
            System.out.println("New
                    Status:" + obj.isAlive());

    }
}
```

2.
```
class A extends Thread {
        public void run()  {
            System.out.println("Status:" +
                            isAlive());
        }
}

public class Test{
    public static void main(String[] args) {
            A obj=new A();
             obj.start();
             try {
                 obj.join();
             }
            catch(Exception e) {}
            System.out.println("New
                    Status:"+obj.isAlive());
     }
}
```

(b) Explain following terms related to Exceptions in java: [4]
1. throw
2. throws
3. IOException
4. ArrayIndexOutofBoundException

(c) Explain Thread synchronization in java using suitable example. [4]