Unique Paper Code	: 32341201
Name of the Course	: B.Sc.(H) Computer Science
Name of the Paper	: Programming in Java (OC)
Semester	: Semester-II
Year of Admission	: 2015, 2016, 2017

Maximum Time : 2 Hours Maximum Marks: 75 Marks

Instructions for Candidates:
Attempt any four questions. All questions carry equal marks.

Q.1 Write a program to define an interface *Matrix* with following methods:

public void addsub(Matrix m1, Matrix m2);
public int calc_trace(Matrix m);

Define class *Myclass* to implement interface *Matrix* with appropriate code for *addsub()* method to add and subtract two matrices, and display the results.
Method *calc_trace()* should calculate the sum of all the diagonal elements for square matrix. In case the input matrix is not square, it should display the error message.

Q.2 Fill appropriate java code in place of blanks.

```
import java.awt.*;
import java.awt._____.*;
import java. _____.*;

/*<_____ code="ME" _____=300 _____=100>
</_____>*/

public class ME extends Applet
implements _____, MouseMotionListener {
String msg = "";
int mouseX = 0, mouseY = 0;

public void init() {
  addMouseListener(this);
  _____(this);
  }

public void _____(MouseEvent me) {
  mouseX = 0;mouseY = 10;
```

```
  msg = "Mouse clicked.";
  repaint();
  }

public void mouseEntered(MouseEvent me) {
  mouseX = 0;
  mouseY = 10;
  msg = "_____";
  repaint(); }

public void _____(MouseEvent me) {
  mouseX = 0;
  mouseY = 10;
  msg = "Mouse exited.";
  repaint(); }

public void mousePressed(MouseEvent me) {
  mouseX = me.getX();
  mouseY = me.getY();
  msg = "Down";
  _____; }

public void _____(MouseEvent me) {
  mouseX = me.getX();
  mouseY = _____;
  msg = "up";
  repaint(); }

public void mouseDragged(MouseEvent me) {
  mouseX = me.getX();
  mouseY = me.getY();
  msg = "*";
  showStatus("Dragging mouse at " + _____ + ", " +
  _____);
  repaint(); }

public void mouseMoved(MouseEvent me) {
  }
public void paint(_____ g) {
  g._____(msg, mouseX, mouseY);
  }
}
```

Q.3 Write Java code to calculate total salary of employee *A* when *HRA*, *DA* and *BasicSalary* is given and calculate total salary of employee *B* when only *DA* and *BasicSalary* is given. Create an abstract class *Salary* with an abstract method *getTotalSalary()*. Class *Salary* is inherited in two classes *AA* and *BB*, each having *getTotalSalary()* as inherited method and returning the total salary of the employee. Pass appropriate values as parameters to the constructors. Create an object for each of the two classes and print the total salary of both the employees. Make use of dynamic method dispatch.

Q.4 Exact output of a multithreaded program cannot be predicted. Explain in detail the working of the following program which uses multithreading. Further, explain why the order of displayed messages differs on separate executions of program. How many threads are executing in the program?

```
public class DisplayMessage implements Runnable {
  private String message;
  public DisplayMessage(String message) {
    this.message = message;
  }
  public void run() {
    while(true) {
      System.out.println(message);
    } }
}
public class ThreadClassDemo {
  public static void main(String [] args) {
    Runnable hello = new DisplayMessage("Hello");
    Thread thread1 = new Thread(hello);
    System.out.println("Starting hello thread...");
    thread1.start();
    Runnable bye = new DisplayMessage("Goodbye");
    Thread thread2 = new Thread(bye);
    thread2.setPriority(Thread.MIN_PRIORITY);
    System.out.println("Starting goodbye thread...");
    thread2.start();
    System.out.println("main() is ending...");
  } }
```

Q.5 Create a user defined exception class which is thrown when number entered by user is not within range of 60 to 90. Override *toString()* method to display error message as "Valid range of numbers to be entered is 60 to 90". Use multi catch feature in the code to catch arithmetic exception also. Why arithmetic exception is an unchecked exception? How is it different from checked exceptions? How can the underlying exception be displayed in Java if required?

Q.6 Make an object of *StringBuffer* class. Initialize it with string "Hello Java". Find length and capacity of this object. Explain the difference in output values. Insert a string "dear" in this object to get output as "Hello dear Java". Make use of predefined methods to convert value of this *StringBuffer* object to "I like Java". Explain how *StringBuffer* class accommodates dynamic strings.