M S RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, affiliated to VTU)
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Term:	22nd Jan – 12th May, 2018	**Course Code:**	IS 62A/B
Course:	OO Analysis and Design Patterns	**Semester:**	6 A, B, C
CIE:	Test – I	**Max Marks:**	30
Date:	11-03-2019	**Time:**	2 – 3 pm

Portions for Test: Lecture Nos. from 01 to 14 as per lesson plan.
Instructions to Candidates: Answer any two full questions. Mobiles are not allowed.

Sl. #	Question	Marks	Bloom's Level #	COs
1a	Explain Martin Fowler's three perspectives of software development with an example design.	8	A	CO1
1b	Define a pattern (both succinctly and detailed). Pattern helps in communicating at different levels simultaneously. Justify this statement with an example design.	7	A	CO2
2	Provide an example for facade or adapter pattern. Draw class diagram, sequence diagrams and brief implementation code and also key features for the same example.	15	C	CO1, CO2
3a	What are the challenges you face while designing an object oriented system using traditional approach? Explain each problem by taking an example design.	8	E	CO1
3b	Compare and contrast agile coding and design patterns.	7	E	CO2

#: A= Analyze, E – Evaluate, C – Create

Scheme of Evaluation:

1a	Martin Fowler's three perspectives of software development: Conceptual, Specification and Implementation An example design.	4 4
1b	Pattern: Succinct defn: Solutin to a problem in a context Detailed: Pattern identifies problem recurring in environment and provides the core of the solution which can be used many times without doing it the same way twice Pattern helps in communicating at different levels simultaneously. Justification	3 4
2	Adapter/facade Pattern Features Example Class diagram Sequence Diagram	5 4 3 3
3a	Problems of traditional OO method: Tight coupling Weak cohesion Class explosion Tall class hierarchies	Any 4(2 each)
3b	agile coding vs design patterns – agile advocates incremental approach, design pattern approach advocates head-on . They both have common goals but different approach. Common Goals:No	7

Scanned with CamScanner

M S Ramaiah Institute of Technology
Department of Information Science and Engineering

Term:	31/01/2019 to 22/05/2019	**Course Code:**	IS63
Course:	Java and J2EE	**Semester:**	6
CIE:	Test-I	**Max Marks:**	30
Date:	12/03/2019	**Time:**	09:30-10:30 AM

Portions for Test: Lecture Nos. from 1 to 16 as per lesson plan

Instructions to Candidates: Answer any **Two** out of Three questions. **Mobile phones are banned**

Q.No	Questions	Marks	Bloom's Level	CO
1a.	Create a User-defined storage class to define an employee class with (id, name, dob, salary, designation) Write appropriate getter and setter methods to store the value to a HashMap Class. Display at least 3 employee details in the ascending order of employee name.	7	Remember	1
1b.	Explain about Java Foundation Classes? Discuss the features of Swings.	8	Remember	2
2a.	List and explain any 5 methods in ArrayList Class along with its constructors. Also write java code to create object of ArrayList of type integer to simulate operations of stack. (Push, Pop, Display)	7	Apply	1
2b.	List any 8 Swing components and explain the features of the components.	8	Remember	2
3a.	Write a java program to create a LinkedList accepting array of Strings. Create a method to identify number of vowels and consonants in each string and display their count. Also write a method to identify duplicate elements in the List.	8	Create	1
3b.	Compare between AWT and Swings. Also discuss Swing packages.	7	Remember	2

Scanned with CamScanner

M S Ramaiah Institute of Technology
Department of Information Science and Engineering

Scheme and Solution

Term:	31/01/2019 to 22/05/2019	**Course Code:**	18S63
Course:	Java and J2EE	**Semester:**	6
CIE:	Test-I	**Max Marks:**	30
Date:	12/03/2019	**Time:**	09:30-10:30AM

Portions for Test: Lecture Nos. from 1 to 16 as per lesson plan
Instructions to Candidates: Answer any **Two** out of Three questions. **Mobile phones are banned**

Q.No	Questions	Marks

1a.	class Employee {
	private long id;
	private String name;
	private Date dateOfBirth;
	private BigDecimal salary;
	}
	
	public static void main(String[] args) {
	HashMap<Employee,String>employeeMap = new HashMap<Employee,String>();
	Employee employee1 = new Employee();
	employee1.setId(1);
	employee1.setName("Sachin");
	employee1.setDateOfBirth(new Date(1987,2,1));
	employee1.setSalary(new BigDecimal(100000));
	
	employeeMap.put(employee1,"India");
	for (Map.Entry<Employee, String>employeeStringEntry : employeeMap.entrySet()) {
	System.out.println(employeeStringEntry.getKey().hashCode());
	}
	
	employee1.setName("Rahul");
	for (Map.Entry<Employee, String>employeeStringEntry : employeeMap.entrySet()) {
	System.out.println(employeeStringEntry.getKey().hashCode());
	}
	
	System.out.println(employeeMap.get(employee1));
	
	Employee employee2 = new Employee();
	employee2.setId(1);
	employee2.setName("Sachin");
	employee2.setDateOfBirth(new Date(1987,2,1));
	employee2.setSalary(new BigDecimal(100000));
	System.out.println(employeeMap.get(employee2));
	}
		7

1b.	JFC Contains: a. Swing b. Cut and paste c. Accessibility features d. The desktop colors features e. Java 2D	3
	JFC Technology a. Accessibility API b. Abstract Windowing Toolkit c. 2D Graphics d. Drag and Drop	3
	Swing Features: a. Lightweight components b. Pluggable Look and Feel	2
2a.		
	Sl. No. / **Method** / **Method and Description**	2

Scanned with CamScanner

1.	boolean **add**(E e)	This method is used to append the specified element to the end of this list.
2.	void **add**(int index, E element)	This method inserts the specified element at the specified position in this list.
3.	boolean **addAll**(Collection<? Extends E> c)	This method appends all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's Iterator.
4.	boolean **addAll**(int index, Collection<? extends E> c)	This method inserts all of the elements in the specified collection into this list, starting at the specified position.
5.	void **clear**()	This method removes all of the elements from this list.
6.	Object **clone**()	This method returns a shallow copy of this ArrayList instance.
7.	boolean **contains**(Object o)	This method returns true if this list contains the specified element.

Stacks using ArrayList

5

```
package stack;

import java.util.ArrayList;
import java.util.EmptyStackException;

public class MyStack<E> {
	private ArrayList<E> al;

	public MyStack() {
		al = new ArrayList<E>();
	}

	public void push(E item) {
		al.add(item);
	}

	public E pop() {
		if (!isEmpty())
			return al.remove(size()-1);
		else
			throw new EmptyStackException();
	}

	public booleanisEmpty() {
		return (al.size() == 0);
	}

	public E peek() {
		if (!isEmpty())
			return al.get(size()-1);
		else
			throw new EmptyStackException();
	}

	public int size() {
		return al.size();
	}

	@Override
	public String toString() {
		return "MyStack [al=" + al.toString() + "]";

	}
}
```

1 mark

2b.

Scanned with CamScanner

JApplet	This class represents Swing applets and hierarchically extends the AWT applet.	each (any 8)
JButton	This class represents Swing buttons. These buttons can possess an icon as well as text with suitable (absolute as well as relative) positioning. You can create a button object with any of the constructors supported as follows: JButtonbuttonRef = new JButton(text, icon); where text and icon are the text label, and the icon is used over the button.	
JCheckBox	This class represents check boxes in Swing	
JComboBox	A Swing combo box is an enhancement over the AWT component called Choice. The combo box is a pull-down list with a capability to display the selected item in a display area. You can also enter an item in the display area to edit it into the list.	
JDialog	This class represents the Swing version of dialog boxes. JDialog extends the AWT dialog box and requires its components to be added to its content pane	
JFileChooser	The class JFileChooser is a dialog box that allows you to open or save information content.	
JFrame	This class represents the Swing frame that is more sophisticated than the AWT frame. You can add components in layers, add a menu bar, or paint over the component (through a component called a glass pane).	
JLabel	This class represents Swing labels that can display a piece of text as well as icons with suitable positioning. A label object can be created by using the following statement: JLabellableRef = new JLabel(text, icon, JLabel.CENTER); This statement creates a label with the specified text and icon and their alignment along the horizontal direction	
JMenu	This class represents Swing menus	
JMenuBar	This class represents the menu bar over a Swing applet or frame.	
JMenuItem	This class represents Swing menu items. Menu items are button-like objects that can trigger specific actions in a program.	
JOptionPane	This class represents the option panes in Swing. Option panes are dialog boxes that display some information to the user or get confirmation from the user so that the program can be executed further based on the choice the user makes.	
JPanel	This class represents Swing panels that have the capability of double buffering along with the ability to group other components.	
JPasswordField	This class represents a text field that does not display the text entered into it, but displays the specified character (such as *) as input is given into the field.	
JPopupMenu	This class represents Swing pop-up menus that contain functionality to be accessed with a mouse click	
JProgressBar	This class represents progress bars in Swing. You can use a progress bar to indicate the state of some operation	
JRadioButton	This class represents Swing radio buttons. By using radio buttons, you can select an option from a given list of mutually exclusive options.	
JRadioButtonMenuItem	Radio button menu items are menu items that represent mutually exclusive items. You can select only one item at a time.	
JTextArea	This class represents text area components in Swing. Text area components allow the editing of multiline plain text.	
JTextPane	This class can be used to create sophisticated editors in Swing applications. A text pane serves as a graphics editor for text as well as pictures.	

3a.

```
class CountVowelsConsonants
{
        LinkedList<String>ll = new LinkedList();
        void countVC()
        {
```

6

Scanned with CamScanner

```
        System.out.println("Enter a character :");
    Scanner sc = new Scanner(System.in);
    char ch = sc.next().charAt(0);
    if(ch == 'a'|| ch == 'e'|| ch == 'i' ||ch == 'o' ||ch == 'u'||ch == ' '){
System.out.println("Given character is an vowel");
    }else{
System.out.println("Given character is a consonant");
    }
        }

        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) {

          /* sequence of steps is important here */
          dup = ptr2.next;
          ptr2.next = ptr2.next.next;
System.gc();
        } else /* This is tricky */ {
          ptr2 = ptr2.next;
        }
      }
      ptr1 = ptr1.next;
    }
  }
}
```

Main method and class — 2

3b. Difference between AWT and Swing — Any 4 (Four marks)

AWT	Swing
AWT stands for Abstract windows toolkit.	Swing is also called as JFC's (Java Foundation classes).
AWT components are called Heavyweight component.	Swings are called light weight component because swing components sits on the top of AWT components and do the work.
AWT components require java.awt package.	Swing components require javax.swing package.
AWT components are platform dependent.	Swing components are made in purely java and they are platform independent.
This feature is not supported in AWT.	We can have different look and feel in Swing.
These feature is not available in AWT.	Swing has many advanced features like JTabel, Jtabbed pane which is not available in AWT. Also Swing components are called "lightweight" because they do not require a native OS object to implement their functionality. JDialog and JFrame are heavyweight, because they do have a peer. So components like JButton, JTextArea, etc., are lightweight because they do not have an OS peer.
With AWT, you have 21 "peers" (one for each control and one for the dialog itself). A "peer" is a widget provided by the operating system, such as a button object or an entry field object.	With Swing, you would have only one peer, the operating system's window object. All of the buttons, entry fields, etc. are drawn by the Swing package on the drawing surface provided by the window object. This is the

Scanned with CamScanner

	reason that Swing has more code. It has to draw the button or other control and implement its behavior instead of relying on the host operating system to perform those functions.	
AWT is a thin layer of code on top of the OS.	Swing is much larger. Swing also has very much richer functionality.	
Using AWT, you have to implement a lot of things yourself.	Swing has them built in.	

Swing packages:
javax.swing
javax.swing.event
javax.swing.plaf.basic
javax.swing.table
javax.swing.border
javax.swing.tree

3 marks

Scanned with CamScanner

M S RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, affiliated to VTU)
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Term:	31st Jan –22nd May, 2019	**Course Code:**	IS 62A/B
Course :	OO Analysis and Design Patterns	**Semester:**	6 A, B, C
CIE:	Test – III	**Max Marks:**	30
Date:	09-05-2019	**Time:**	2 – 3 pm

Portions for Test: Lecture Nos. from 31 to 42 as per lesson plan.
Instructions to Candidates: Answer any two full questions. Mobiles are not allowed.

Sl. #	Question	Marks	BL#	COs
1a.	Explain the three categories of design patterns? Is there a fourth category? If yes, state its purpose.	7	A	CO4
b.	How do factories increase cohesion, decrease coupling and assist in testing?	8	A	CO5
2.	The steps to prepare coffee or tea are same, say, boil water, brew coffee/tea grinds, add condiments, and serve. But how some steps are performed is different like you may add milk as condiment to coffee and lemon as condiment to tea. Which pattern is suitable to implement this problem? Write class diagram and give either C++ or Java code for the same design	7	E	CO4
b.	Is singleton pattern safe in multithreaded environment? If not, give reasons and some solutions to make it thread safe.	8	U	CO5
3a.	In the International E-commerce case study, if we want to send welcome email to new customer and offer notifications as and when required, which design pattern can be applied so that system and customer objects are independent from each other? Write the class diagram and code in java/C++ for the same.	7	E	CO4
b.	Explain factory method pattern with an example application.	8	U	CO5

#: U – Understand, A= Analyze, E – Evaluate

Scheme

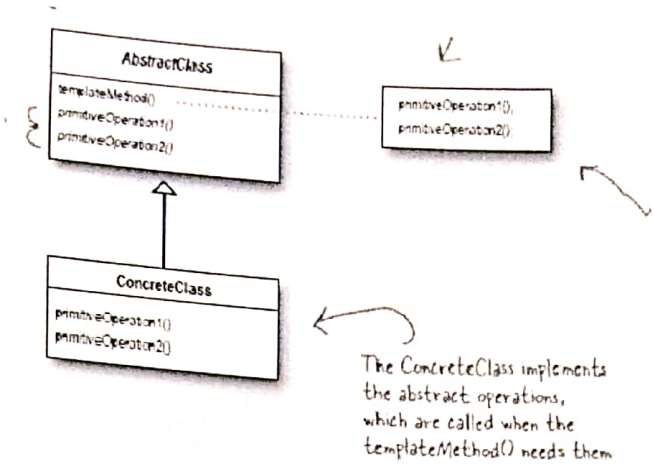
1a. Creational, Behavioral and Structural Patterns. 4

Yes, there is a fourth category called Decoupled. Example pattern is Observer 3

b. Factories separate object creation taking the burden out of object users. So one class one responsibility which means increased cohesion. When new types of object are created, users are decoupled from the logic of creation. So increased coupling. Users of object work with all types of objects in the same manners. So less testing. 8

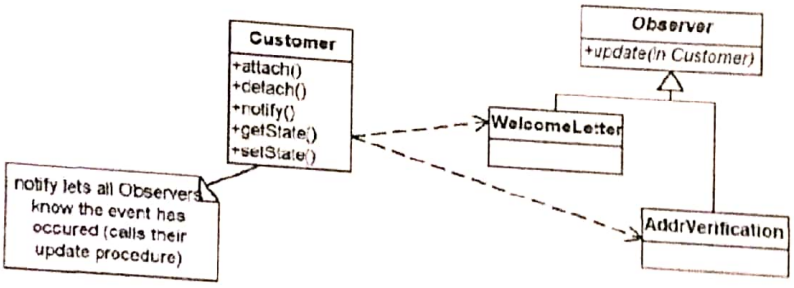
2a. Template method pattern is most suitable. 7

Scanned with CamScanner

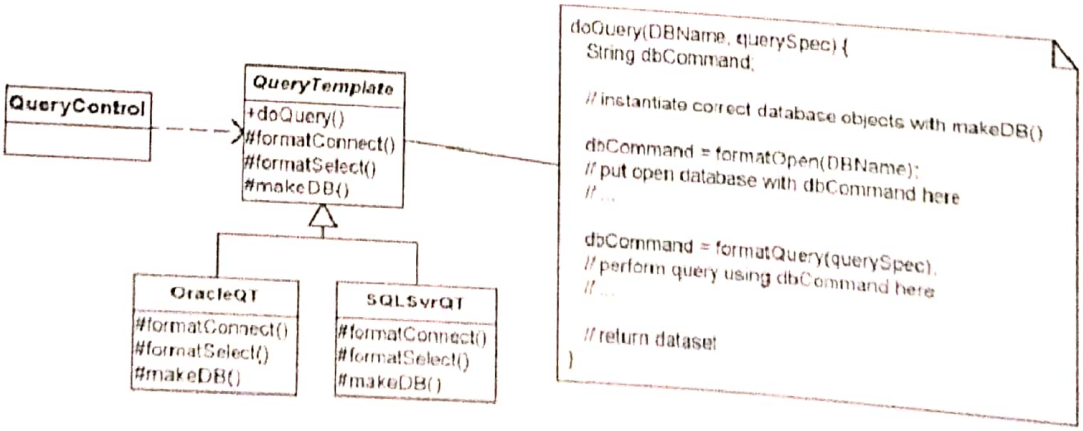


b. No, singleton is not thread safe. It can be corrected by synchronization or using double checked locking pattern. 8

3a. Observer Pattern:



b. Factory method pattern: 8



Scanned with CamScanner

M S Ramaiah Institute of Technology
Department of Information Science and Engineering

Term:	31/01/2019 to 22/05/2019	**Course Code:**	IS63
Course:	Java and J2EE	**Semester:**	6
CIE:	Test-II	**Max Marks:**	30
Date:	09/04/2019	**Time:**	09:30-10:30 AM

Portions for Test: Lecture Nos. from 17 to 23 as per lesson plan
Instructions to Candidates: Answer any **Two** out of Three questions. **Mobile phones are banned**

Q.No	Questions	Marks	Bloom's Level	CO
1a.	Write a program using swings to read purchase details of a customer such as customer number, customer id, items purchased(select a set of items from a list of items) . On clicking of the submit button display the list of items purchased by the customer and the total number of items purchased into a text area.	8	Create	2
1b.	Discuss the life cycle of a servlet with a diagram.	7	Understand	3
2a.	Explain the components of the root pane class.	7	Understand	2
2b.	Write a html code that allows student name entry, student reg number, student subject(options) and department(option) . Define a servlet that retrieves the details entered and inserts these details into a table student of database studentdetails. Also print all the details of student table.	8	Create	3
3a.	Using java swings, read data such as email address and password through input dialog boxes. Place the read data into text boxes. Check for validity of the email address and password and pop up appropriate message dialog boxes if mismatch occurs. Generate an option dialog box that allows the user to indicate the type of email address(personal, business, official). Place the type of selected email address into another text box.	8	Apply	2
3b.	What are the different ways of executing queries in JDBC? Explain any 2 with an example.	7	Analyze	3

Scanned with CamScanner

M S Ramaiah Institute of Technology
Department of Information Science and Engineering

Term:	31/01/2019 to 22/05/2019	**Course Code:**	IS63
Course:	Java and J2EE	**Semester:**	6
CIE:	Test-II	**Max Marks:**	30
Date:	09/04/2019	**Time:**	09:30-10:30 AM

Portions for Test: Lecture Nos. from 17 to 23 as per lesson plan
Instructions to Candidates: Answer any **Two** out of Three questions. **Mobile phones are banned**

Q.No	Questions	Marks

1a.	Write a program using swings to read purchase details of a customer such as customer number, customer id, items purchased(select a set of items from a list of items) . On clicking of the submit button display the list of items purchased by the customer and the total number of items purchased into a text area.
	
	public class Personal extends JFrame implements ActionListener
	{
	JLabel cust_n = new JLabel("Enter Cust number");
	JLabel cust_id = new JLabel("Enter Cust id");
	JTextField custt = new JTextField(20);
	JTextField custid= new JTextField(20);
	// create check boxes based on number of items
	i.e JCheckBox jcbox1 = new JCheckBox("Item1");
	JCheckBox jcbox2 = new JCheckBox("Item2");
	JButton btn1 = new JButton("Submit");
	JTextArea tarea = new JTextArea(30,30);
		2 M

	Personal()
	{
	//add swing components to frame
	btn1.addActionListener(this);
	}
		2 M

	public void actionPerformed(ActionEvent e)
	{int n=0
	tarea.append(custt.getText());
	tarea.append("with" + custid.getText());
	if(jcbox1.isSelected() == true)
	{ n++;
	tarea.append("Items selected" + jcbox1.getText());
	}
	if(jcbox2.isSelected() == true)
	{
	n++;
	tarea.append("Items selected" + jcbox2.getText());
	}
	tarea.append("Number of items"+n);
	}
		3 M

	// write main program and create object of Personal class }	1 M
1b.	Discuss the life cycle of a servlet with a diagram. 1. Server loads the servlet	7 M

Scanned with CamScanner

2. Server creates one or more instances of the Servlet class
3. Server calls the init() method of each servlet instance
4. Servlet request is received
5. Servlet selects the Servlet instances
6. Servlet calls the service() method of the Servlet instance
7. Servlet service() method processes the request
8. Servlet service() method then returns the output to the client
9. Servlet wait until next request is received and it is unloaded by the server
10. Server unloads the Servlet after calling its destroy() method.

2a. Explain the components of the root pane class.

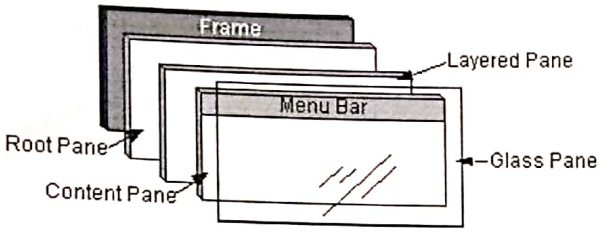


Diagram — 3 M

Explanation — 4 M

2b. Write a html code that allows student name entry, student reg number, student subject(options) and department(option) . Define a servlet that retrieves the details entered and inserts these details into a table student of database studentdetails. Also print all the details of student table.

HTML file	2 M
servlet program	
read parameters	1 M
Connect to Database	2 M
insert	1 M
select to read table	2 M

In service(), toGet() or toPost()
Use getParameter() to retrieve contents of HTML files
// Connect to database

```
try {
        // Register JDBC driver
        Class.forName("com.mysql.jdbc.Driver");

        // Open a connection
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:8806/student",
"root", "root");
// Execute SQL query
 Statement stmt = conn.createStatement();
 String query1 = "insert into studentdetails ….."// complete with read
parameters
stmt.executeUpdate(query1);
String query2 = "select * from studentdetails;"
ResultSet rs = stmt.executeQuery(query2)
```

Scanned with CamScanner

	while(rs.hasNext())
	{
	//read values and print on thru the PrintWriter object
	}
	catch(SQLException e)
	{
	}	

3a.	Using java swings, read data such as email address and password through input dialog boxes. Place the read data into text boxes. Check for validity of the email address and password and pop up appropriate message dialog boxes if mismatch occurs. Generate an option dialog box that allows the user to indicate the type of email address(personal, business, official). Place the type of selected email address into another text box.
	
	// Read data in from input dialog box using JOptionPane.showInputDialog() and place it into a textbox
	
	i.e JTextField jtf1 = new TextField(JOptionPane.showInputDialog(null,"enter email"));
	
	//retrieve the data from text box and check for validity
	
	if(jtf1.getText().equals(abc@gmail.com))
	
	JOptionPane.showMessageDialog(null, "Valid email");
	
	else
	
	JOptionPane.showMessageDialog(null, "InValid email");
	
	//generate an option dialog box
	
	String choice[] = {"personal","business","official");
	
	showOptionDialog(null, "Click on a choice", "Click a button", JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, null, choice, choice[0]);
	
	Main Program + complete program
		2 M
		
		2 M
		
		2 M
		
		2 M

3b.	What are the different ways of executing queries in JDBC? Explain any 2 with an example. Different ways of executing queries in JDBC: • `Statement`: Used to implement simple SQL statements with no parameters. • `PreparedStatement`: Used for precompiling SQL statements that might contain input parameters. • `CallableStatement`: Used to execute stored procedures that may contain both input and output parameters. Explain any 2 with example:	3 M 2 X 2 M

Scanned with CamScanner

M S Ramaiah Institute of Technology
Department of Information Science and Engineering

Term:	31/01/2019 to 22/05/2019	**Course Code:**	IS63
Course:	Java and J2EE	**Semester:**	6
CIE:	Test-III	**Max Marks:**	30
Date:	10/05/2019	**Time:**	09:30-10:30 AM

Portions for Test: Lecture Nos. from 1 to 16 as per lesson plan
Instructions to Candidates: Answer any **Two** out of Three questions. **Mobile phones are banned**

Q.No	Questions	Marks	Bloom's Level	CO
1a.	Write a JSP program to create a shopping cart for clothes. Fetch the details of clothes like type, brand and cost from database. Calculate the total amount and print the bill on a HTML page with all the details.	8	Apply	4
1b.	Explain any seven methods defined in the class DatagramPacket with its prototype.	7	Remember	5
2a.	Explain about Factory Methods and Instance Methods in InetAddress.	8	Remember	4
2b.	Write a Java Bean program to convert Currency from USD to INR.	7	Understand	4
3a.	Write a program in java.net to print whois information about a domain name like: Domain Name, Registrar, Whois Server, Referral URL, Nameservers. Also print information like Protocol, Port, Host, File and External Form of the same URL.	8	Apply	5
3b.	Explain in detail the features / benefits of Enterprise Bean	7	Remember	4

Scanned with CamScanner

M S Ramaiah Institute of Technology
Department of Information Science and Engineering
Scheme and Solution

Term:	31/01/2019 to 22/05/2019	**Course Code:**	IS63
Course:	Java and J2EE	**Semester:**	6
CIE:	Test-III	**Max Marks:**	30
Date:	10/05/2019	**Time:**	09:30-10:30 AM

Portions for Test: Lecture Nos. from 1 to 16 as per lesson plan

Instructions to Candidates: Answer any **Two** out of Three questions. **Mobile phones are banned**

Q.No	Questions	Marks
1a.	Write a JSP program to create a shopping cart for clothes. Fetch the details of clothes like type, brand and cost from database. Calculate the total amount and print the bill on a HTML page with all the details. JSP Page to display Items:	4

```
//File: carts.html
<html>

<head>
   <title>carts</title>
</head>

 <body bgcolor="white">
<font size = 5 color="#CC0000">

<form type=POST action=carts.jsp>
<BR>
Please enter item to add or remove:
<br>
Add Item:

<SELECT NAME="item">
<OPTION>T-Shirt</OPTION>
<OPTION>Formal Shirt</OPTION>
<OPTION>PullOver</OPTION>
<OPTION>T-Shirt round neck</OPTION>
<OPTION>Casual Shirt</OPTION>
<OPTION>Trousers</OPTION>
<OPTION>Jeans</OPTION>
<OPTION>Belt</OPTION>
<OPTION>Caps</OPTION>
</SELECT>


<br> <br>
<INPUT TYPE=submit name="submit" value="add">
<INPUT TYPE=submit name="submit" value="remove">

</form>

</FONT>
</body>
</html>
```

//

4

```
//File: carts.jsp

<html>

<jsp:useBean id="cart" scope="session" class="sessions.DummyCart" />

<jsp:setProperty name="cart" property="*" />
<%
 cart.processRequest(request);
%>


<FONT size = 5 COLOR="#CC0000">
<br> You have the following items in your cart:
<ol>
<%
 String[] items = cart.getItems();
 for (int i=0; i<items.length; i++) {
%>
<li> <% out.print(util.HTMLFilter.filter(items[i])); %>
```

Scanned with CamScanner

```
<%
}
%>
</ol>

</FONT>

<hr>
<%@ include file ="carts.html" %>
</html>
```

1b. Any seven methods defined in the class DatagramPacket with its prototype. — 1 mark each any 7

InetAddress getAddress()	Returns the IP address of the machine to which this datagram is being sent or from which the datagram was received.
byte[]**getData**()	Returns the data buffer.
int **getLength**()	Returns the length of the data to be sent or the length of the data received.
int **getOffset**()	Returns the offset of the data to be sent or the offset of the data received.
int **getPort**()	Returns the port number on the remote host to which this datagram is being sent or from which the datagram was received.
SocketAddress getSocketAddress()	Gets the SocketAddress (usually IP address + port number) of the remote host that this packet is being sent to or is coming from.
void **setAddress**(**InetAddress** iaddr)	Sets the IP address of the machine to which this datagram is being sent.
void **setData**(byte[] buf)	Set the data buffer for this packet.
void **setData**(byte[] buf, int offset, int length)	Set the data buffer for this packet.
Void **setLength**(int length)	Set the length for this packet.
Void **setPort**(int iport)	Sets the port number on the remote host to which this datagram is being sent.
Void **setSocketAddress**(**SocketAddress** address)	Sets the SocketAddress (usually IP address + port number) of the remote host to which this datagram is being sent.

2a. Explain about Factory Methods and Instance Methods in InetAddress. — 8

Factory methods:

Method	Description
public static InetAddress getByName(String host) throws UnknownHostException	it returns the instance of InetAddress containing LocalHost IP and name.
public static InetAddress	it returns the instance of InetAdddress

Scanned with CamScanner

getLocalHost() throws UnknownHostException	containing local host name and address.
public String getHostName()	it returns the host name of the IP address.
public String getHostAddress()	it returns the IP address in string format.

Instance Methods

Boolean equals(Object obj)	Compares this object against the specified object
byte[]getAddress()	Returns the raw IP address of this InetAddress object
String getHostAddress()	Returns the IP address string in textual presentation
String getHostName()	Gets the host name for this IP address
Boolean isMulticastAddress()	Utility routine to check if the InetAddress is an IP multicast address
String toString()	Converts this IP address to a String

2b. Write a Java Bean program to convert Currency from USD to INR. 4

```
package ejb;

import java.math.BigDecimal;
import javax.ejb.Stateless;

@Stateless

public clas CurrencyConverterBean implements ejb.CurrencyConverter
{
	private BigDecimal USD = new BigDecimal("0.0229137");
	private BigDecimal INR = new BigDecimal("46.589100");

	public BigDecimal convert(String from, String to, BigDecimal amount)
	{
		if(from.equals(to))
		{
			return amount;
		}
		else
		{
			BigDecimal toRate = findRate(to);
			BigDecimal result = toRate.multiply(amount);
			return result.setScale(2, BigDecimal.ROUND_UP);
		}
	}

	public BigDecimal findRate(String to)
	{
		BigDecimal returnValue = null;

		if(to.equals("INR")
		{
			returnValue = INR;
		}
		if(to.equals("USD")
		{
			returnValue = USD;
		}
		return returnValue;
	}
}
```

3

HTML page

```
<%@ page import="ejb.CurrencyConverter, java.math.BigDecimal, javax.naming.InitialContext"
%>

<%!
	private CurrencyConverter converter = null;

	public void jspInit()
	{
		try
		{
			InitialContext ic = new InitialContext();

			converter = (CurrencyConverter)
```

Scanned with CamScanner

	ic.lookup(CurrencyConverter.class.getName());
	}
	catch(Exception ex)
	{
	}
	}
	
	public void jspDestroy()
	{
	converter = null;
	}
	%>
	
	HTML Code
	
	<%
	String amount = request.getParameter("amount");
	if(amount != null && amount.length() >0)
	{
	BigDecimal d = new BigDecimal(amount);
	BigDecimal convertedAmount = converter.convert(request.getParameter("From"), request.getParameter("To"), d);
	}
	%>	

3a.	Socket s = new Socket("whois.internic.net", 43); Display Whois information URL hp = new URL("www.google.com/abc.txt"); Hp.getProtocol() Hp.getPort() Hp.getHost() Hp.getFile() Hp.toExternalForm()	4 4
3b.	Explain in detail the features / benefits of Enterprise Bean 1. Complete focus only on business logic 2. Reusable Components 3. Portable 4. Fast Building of Applications 5. One Business Logic having many presentation logics 6. Distributed Deployment 7. Application Interoperability 8. Customized Deployment	1 mark each any 7

Scanned with CamScanner