Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058, India

(Autonomous College Affiliated to University of Mumbai)

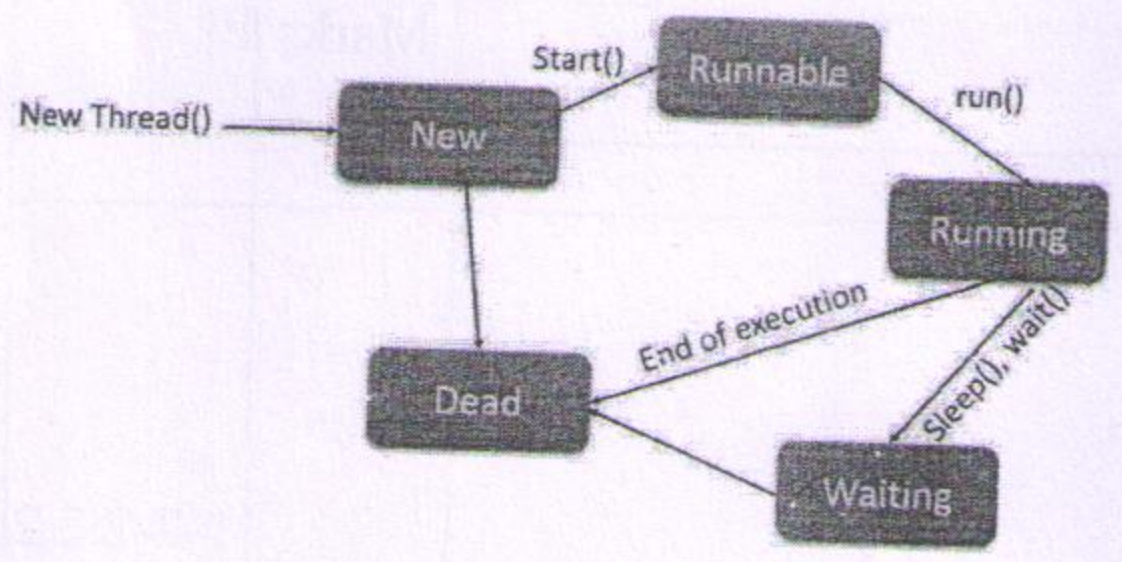
End Semester Examination

Answer Key Nov 2019

Max. Marks: 60	Duration: 3 Hrs
Class: S.Y.	Semester: III
Course Code: MCA 31	Branch: M.C.A.
Name of the Course: Core and Advanced JAVA	

Instruction:
(1) All questions are compulsory
(2) Draw neat diagrams
(3) Assume suitable data if necessary

Q No.		Max. Marks	CO-BL-PI
Q. 1	Attempt any Four	12	
A.	Each correct ans ½ marks Long Form • JVM – JAVA virtual Machine • JRE – JAVA runtime environment • AWT – Abstract window toolkit • JDBC – JAVA database connectivity • JSP – JAVA server side pages • AOP – Aspect Oriented programming		1-2-2.2.2
B.	Each advantages – 1 mark Advantage of Java Package 1) Java package is used to categorize the classes and interfaces so that they can be easily maintained. 2) Java package provides access protection. 3) Java package removes naming collision.		1-2-3.1.3
C.	Three step – each step – 1mark Working of JVM The JVM can contain the program and prevent it from generating side effects outside of the system. The use of bytecode enables the Java run-time system to execute programs much faster than you might expect. When the JIT compiler is part of the JVM, it compiles bytecode into executable code in real time, on a piece-by-piece, demand basis. It is important to understand that it is		1-2-2.2.2

	not possible to compile an entire Java program into executable code all at once, because Java performs various run-time checks that can be done only at run time. Instead, the JIT compiles code as it is needed, during execution. However, the just-in-time approach still yields a significant performance boost.		
D.	Each difference point – 1 mark Difference between scanner and bufferReader • Scanner is used for parsing tokens from the contents of the stream while BufferedReader just reads the stream and does not do any special parsing. • BufferedReader is synchronized and Scanner is not, Use BufferedReader if you're working with multiple threads. • Scanner hides IOException while BufferedReader throws it immediately. • the Scanner has a smaller buffer (1024 chars) as opposed to the BufferedReader (8192 chars), but it's more than sufficient. • A scanner however is not thread safe, it has to be externally synchronized		1-2-2.2.5
E	Diagram – 1 mark Working of thread – 2 marks  New Thread() New Start() Runnable run() Running End of execution Sleep(), wait() Dead Waiting Each part of such a program is called a thread, and each thread defines a separate path of execution. Thus, multithreading is a specialized form of multitasking. A thread can be running. It can be ready to run as soon as it gets CPU time. A running thread can be suspended, which temporarily suspends its activity. A suspended thread can then be resumed, allowing it to pick up where it left off. A thread can be blocked when waiting for a resource. At any time, a thread can be terminated, which halts its execution immediately. Once terminated, a thread cannot be resumed.		1-2-2.2.4
F.	Each difference – 1mark Differences between implementing Runnable interface and extending Thread class - 1. Multiple inheritance in not allowed in java : When we implement Runnable interface we can extend another class as well, but if we extend Thread class we cannot extend any other class because java does not allow multiple inheritance. So, same work is done by implementing Runnable and extending Thread but in case of implementing Runnable we are still left with option of extending some other class. So, it's better to implement Runnable. 2. Thread safety : When we implement Runnable interface, same object is shared amongst multiple threads, but when we extend Thread class each and every thread gets associated with new object. 3. Inheritance (Implementing Runnable is lightweight operation) : When we extend Thread		1-3-1.3.1

	unnecessary all Thread class features are inherited, but when we implement Runnable interface no extra feature are inherited, as Runnable only consists only of one abstract method i.e. run() method. So, implementing Runnable is lightweight operation. 4. Coding to interface : Even java recommends coding to interface. So, we must implement Runnable rather than extending thread. Also, Thread class implements Runnable interface. 5. Don't extend unless you wanna modify fundamental behaviour of class, Runnable interface has only one abstract method i.e. run() : We must extend Thread only when you are looking to modify run() and other methods as well. If you are simply looking to modify only the run() method implementing Runnable is the best option (Runnable interface has only one abstract method i.e. run()). We must not extend Thread class unless we're looking to modify fundamental behaviour of Thread class. 6. Flexibility in code when we implement Runnable : When we extend Thread first a fall all thread features are inherited and our class becomes direct subclass of Thread , so whatever action we are doing is in Thread class. But, when we implement Runnable we create a new thread and pass runnable object as parameter, we could pass runnable object to executorService & much more. So, we have more options when we implement Runnable and our code becomes more flexible. 7. ExecutorService : If we implement Runnable, we can start multiple thread created on runnable object with ExecutorService (because we can start Runnable object with new threads), but not in the case when we extend Thread (because thread can be started only once).		

Q. 2
A
	Bounded type – 1 mark
	Explanation – 4 marks
	Substitution principal – 1 mark
	
	the type parameters could be replaced by any class type the superclass from which all type arguments must be derived. This is accomplished through the use of an extends clause when specifying the type parameter, as shown here: <T extends superclass> This specifies that T can only be replaced by superclass, or subclasses of superclass. Thus, superclass defines an inclusive, upper limit.
	
	class Stats<T extends Number> { Because the type T is now bounded by Number, the Java compiler knows that all objects of type T can call doubleValue( ) because it is a method declared by Number.
	
	When a bound includes an interface type, only type arguments that implement that interface are legal. When specifying a bound that has a class and an interface, or multiple interfaces, use the & operator to connect them. □ E – Element (used by the Java Collections Framework, for example ArrayList, Set etc.) □ K – Key (Used in Map) □ N – Number □ T – Type □ V – Value (Used in Map) □ S,U,V etc. – 2nd, 3rd, 4th types
	
	Substitution Principle: a variable of a given type may be assigned a value of any subtype of that type, and a method with a parameter of a given type may be invoked with an argument of any subtype of that type.
	interface Collection<E> {
	public boolean add(E elt);
	...
	}
	According to the Substitution Principle, if we have a collection of numbers, we may
		06	2-2-2.2.2

add an integer or a double to it, because Integer and Double are subtypes of Number.
List<Number> nums = new ArrayList<Number>();

2-2-2.1.2

OR
Each Statement explanation – 2 marks => 2*3 = 6 marks

There are three different kinds of statements

1. Statement: Used to implement simple SQL statements with no parameters. You execute Statement objects, and they generate ResultSet objects, which is a table of data representing a database result set. You need a Connection object to create a Statement object.
2. Create statement The createStatement() method of Connection interface is used to create statement. The object of statement is responsible to execute queries with the database.
   public Statement createStatement()throws SQLException Statement stmt = con.createStatement();
3. PreparedStatement: (Extends Statement.) Used for precompiling SQL statements that might contain input parameters. This statement gives you the flexibility of supplying arguments dynamically. String SQL = "Update Employees SET age = ? WHERE id = ?"; pstmt = conn.prepareStatement(SQL);

Q. 2 B

Each type with explanation – 2 marks => 2*3 = 6 marks

The possible RSType are given below. If you do not specify any ResultSet type, you will automatically get one that is TYPE_FORWARD_ONLY.

Type	Description
ResultSet.TYPE_FORWARD_ONLY	The cursor can only move forward in the result set.
ResultSet.TYPE_SCROLL_INSENSITIVE	The cursor can scroll forward and backward, and the result set is not sensitive to changes made by others to the database that occur after the result set was created.
ResultSet.TYPE_SCROLL_SENSITIVE.	The cursor can scroll forward and backward, and the result set is sensitive to changes made by others

06 2-3-2.1.2

To execute a query, call an execute method from Statement such as the following:

1. execute: Returns true if the first object that the query returns is a ResultSet object. Use this method if the query could return one or more ResultSet objects. Retrieve the ResultSet objects returned from the query by repeatedly calling Statement.getResultSet.
2. executeQuery: The executeQuery() method of Statement interface is used to execute queries to the database. This method returns the object of ResultSet that can be used to get all the records of a table.
3. executeUpdate: Returns an integer representing the number of rows affected by the SQL statement. Use this method if you are using INSERT, DELETE, or UPDATE SQL statements.

Q. 3 A	Each comparison – 1 mark => 1*6 = 6 marks
	
	Comparison between
	• BeanFactory is also called basic IOC and ApplicationContext is called Advanced IOC.
	• BeanFactory uses lazy initialization approach whereas ApplicationContext uses eager initialization approach. i.e BeanFactory creates a singleton bean only when it is requested from it but ApplicationContext creates all singleton beans at the time of its own initialization.
	• ApplicationContext creates and manages resources objects on its own whereas BeanFactory used to be explicitly provided a resource object using the syntax :
	ClassPathResource resource = new ClassPathResource("beans.xml");
	XmlBeanFactory factory = new XmlBeanFactory(resource); // Here resource object is provided explicitly
	• ApplicationContext supports internationalization but BeanFactory do not.
	• Annotation based dependency Injection is not supported by BeanFactory whereas ApplicationContext supports using annotation @PreDestroy, @Autowired.
		06	4-3-2.2.5

Q. 3 B.	Each Element – 2 marks = 2*3 = 6 marks Core elements of Hibernate 1. hibernate.cfg.xml: This file has database connection details hbm.xml or Annotation: Defines the database table mapping with POJO. Also defines the relation between tables in java way. 2. Session Factory: There will be a session factory per database. The SessionFacory is built once at start-up It is a thread safe class SessionFactory will create a new Session object when requested 3. Session: The Session object will get physical connection to the database. Session is the Java object used for any DB operations. Session is not thread safe. Hence do not share hibernate session between threads Session represents unit of work with database Session should be closed once the task is completed OR Each Element – 2 marks = 2*3 = 6 marks In Hibernate, an object can remain in three states • Transient: Newly created instance of a persistence class which is never associated with any hibernate session. • Persistence: An object which is associated with the hibernate session is called Persistence object. Any change in the object will reflect in database upon the flash strategy i.e. • automatic flush whenever any property of the persistence object changes • explicit flush by calling session.flush() • Detached: The object which was associated with session earlier but currently not associated with it are called detached object. We can reattach the detached object by calling session.update() or session.saveOrUpdate()method. Once the session will be closed then the same object will be detached again.	06	4-2-2.3.2

Q. 4 A.

Each step with diagram – 3 marks
Comparison between generic and http – 3 marks

Servlet Application works

- Web container is responsible for managing execution of servlets and JSP pages for Java EE application.
- When a request comes in for a servlet, the server hands the request to the Web Container.
- Web Container is responsible for instantiating the servlet or creating a new thread to handle the request.
- Its the job of Web Container to get the request and response to the servlet. The container creates multiple threads to process multiple requests to a single servlet.
- Servlets don't have a main() method. Web Container manages the life cycle of a Servlet instance.

GENERICSERVLET	HTTPSERVLET
Can be used with any protocol (means, can handle any protocol). Protocol independent.	Should be used with HTTP protocol only (can handle HTTP specific protocols) . Protocol dependent.
All methods are concrete except service() method. service() method is abstract method.	All methods are concrete (non-abstract). service() is non-abstract method.
service() should be overridden being abstract in super interface.	service() method need not be overridden.
It is a must to use service() method as it is a callback method.	Being service() is non-abstract, it can be replaced by doGet() or doPost() methods.
Extends Object and implements interfaces Servlet, ServletConfig and Serializable.	Extends GenericServlet and implements interface Serializable

o

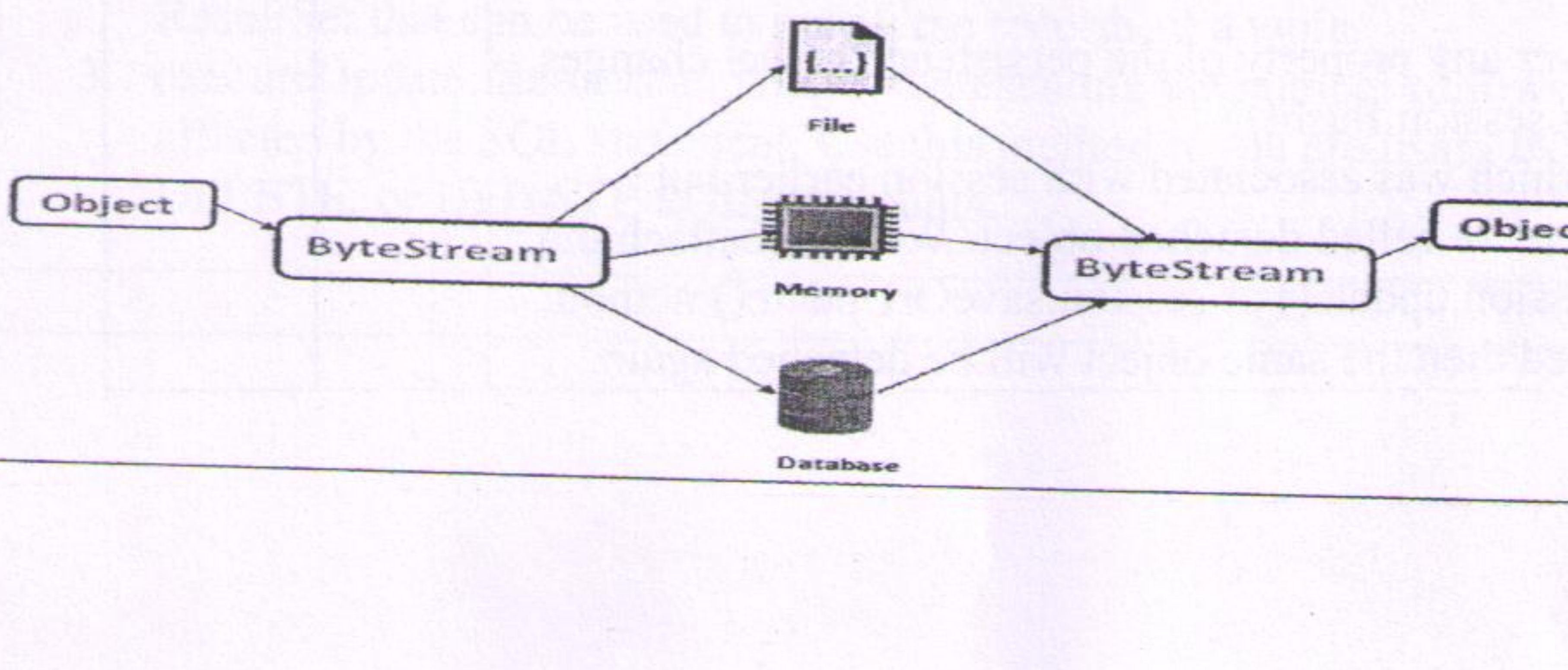
06 3-2-1.3.1

Q. 4 B

Definition of Serialization – 2 marks
Working of serialization – 2 marks
Advantages of serialization – 2 marks

Serialization is a mechanism of converting the state of an object into a byte stream. Deserialization is the reverse process where the byte stream is used to recreate the actual Java object in memory. This mechanism is used to persist the object.

06 3-2-2.2.5



The byte stream created is platform independent. So, the object serialized on one platform can be deserialized on a different platform.
To make a Java object serializable we implement the java.io.Serializable interface.
The ObjectOutputStream class contains writeObject() method for serializing an Object.
public final void writeObject(Object obj) throws IOException
The ObjectInputStream class contains readObject() method for deserializing an object.
public final Object readObject() throws IOException, ClassNotFoundException

Advantages of Serialization
1. To save/persist state of an object.
2. To travel an object across a network.

Only the objects of those classes can be serialized which are implementing java.io.Serializableinterface. Serializable is a marker interface (has no data member and method). It is used to "mark" java classes so that objects of these classes may get certain capability.

Q. 5A

JSP processing – 2 marks
Four phases of JSP – 4 marks

JSP Processing

- As with a normal page, your browser sends an HTTP request to the web server.
- The web server recognizes that the HTTP request is for a JSP page and forwards it to a JSP engine. This is done by using the URL or JSP page which ends with .jsp instead of .html.
- The JSP engine loads the JSP page from disk and converts it into a servlet content. This conversion is very simple in which all template text is converted to println( ) statements and all JSP elements are converted to Java code. This code implements the corresponding dynamic behavior of the page.
- The JSP engine compiles the servlet into an executable class and forwards the original request to a servlet engine.
- A part of the web server called the servlet engine loads the Servlet class and executes it. During execution, the servlet produces an output in HTML format. The output is further passed on to the web server by the servlet engine inside an HTTP response.
- The web server forwards the HTTP response to your browser in terms of static HTML content.
- Finally, the web browser handles the dynamically-generated HTML page inside the HTTP response exactly as if it were a static page.

The following are the paths followed by a JSP –

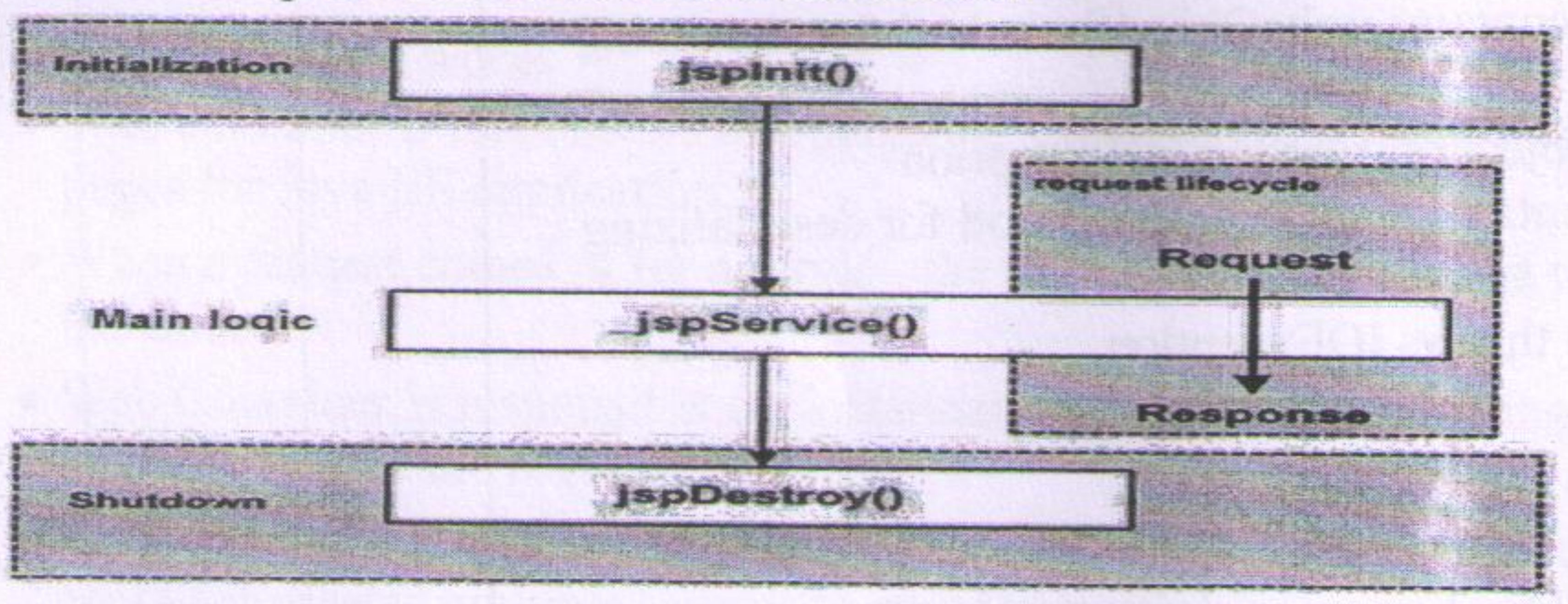
- Compilation
- Initialization
- Execution

06

3-3-1.3.1

- Cleanup

The four phases have been described below –



Q. 5 B

Correct login – 2marks
Correct syntax – 4 marks

```
index.html
<form method="post" action="Validate">
Name:<input type="text" name="user" /><br/>
Password:<input type="password" name="pass" ><br/>
<input type="submit" value="submit">
</form>

Validate.java
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Validate extends HttpServlet {
protected void doPost (HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
  response.setContentType("text/html;charset=UTF-8");
  PrintWriter out = response.getWriter();
  try {
    String name = request.getParameter("user");
    String password = request.getParameter("pass");

    if(password.equals("studytonight"))
    {
      Cookie ck = new Cookie("username",name);
      response.addCookie(ck);
      response.sendRedirect("First");        }
    else
    {
     out.println("<font color='red'><b>You have entered incorrect passw
ord</b></font>");
     RequestDispatcher rd = request.getRequestDispatcher("index.html");
     rd.include(request, response);
       }
```

06 3-4-4.2.1