**1) What is J2EE?**

J2EE means Java 2 Enterprise Edition. The functionality of J2EE is developing multitier web-based applications .The J2EE platform is consists of a set of services, application programming interfaces (APIs), and protocols.

**2) What are the four components of J2EE application?**

* Application client’s components.
* Servlet and JSP technology are web components.
* Business components (JavaBeans).
* Resource adapter components

**3) What are types of J2EE clients?**

* Applets
* Application clients
* Java Web Start-enabled clients, by Java Web Start technology.
* Wireless clients, based on MIDP technology.

**4) What are considered as a web component?**

Java Servlet and Java Server Pages technology components are web components. Servlets are Java programming language that dynamically receive requests and make responses. JSP pages execute as servlets but allow a more natural approach to creating static content.

**5)** **Differentiate between .ear,  .jar and .war files?**

**.jar files:** These files are with the .jar extenstion. The .jar files contains the libraries, resources and accessories files like property files.  
**.war files:** These files are with the .war extension. The .war file contains jsp, html, javascript and other files for necessary for the development of web applications.  
**.ear files:** The .ear file contains the EJB modules of the application.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **JAR** | **WAR** | **EAR** |
| File name | Java Archive File. | Web Archive File. | Enterprise Archive File. |
| Functions | Used to package classes and property files. | Used to package a web application. | Used to package EJB, JSP, Servlets etc. |
| Extension | .jar | .war | .ear |
| Files contain | Enterprise Java Beans (EJB) class files and EJB deployment descriptor. | Servlet class files, JSP files, supporting files, GIF and HTML. | All files of JAR and WAR. |

**6)** **Explain JSP and tell its uses?**

JSP stands for Java Server Pages. It is a presentation layer technology independent of platform. It comes with SUN’s J2EE platforms. They are like HTML pages but with Java code pieces embedded in them. They are saved with a .jsp extension. They are compiled using JSP compiler in the background and generate a Servlet from the page.

**7)** **Explain implicit objects in JSP?**

Objects created by web container and contain information regarding a particular request, application or page are called Implicit Objects. They are:

1) response 2) exception 3)application 4)request

5)session 6)page 7)out 8)config 9)pageContext

**8) Differentiate between response.sendRedirect(url) and <jsp:forward page => ?**

<jsp.forward> element forwards the request object from 1 JSP file to another. Target file can be HTML, servlet or another JSP file, but it should be in the same application context as forwarding JSP file.

sendRedirect send HTTP temporary redirect response to the browser. The browser then creates a new request for the redirected page. It kills the session variables.

**9)** **How can a thread safe JSP page be implemented?**

It can be done by having them implemented by the SingleThreadModel Interface. Add <%@page isThreadSafe=”false” %> directive in the JSP page.

**10)** **What are JSP Actions?**

They are XML tags, which direct the server to using existing components or control behavior of JSP Engine. They consist of a typical prefix of “jsp:” and action name.

<jsp:include/> <jsp:getProperty/>

<jsp:forward/> <jsp:setProperty/>

<jsp:usebean/> <jsp:plugin/>

**11)** **Differentiate between <jsp:include page=…> and <%@include file=>?**

Both these tags include information from 1 page to another.

The first tag acts as a function call between two Jsp’s. It is executed each time client page is accessed by the client. It is useful to modularize the web application. New content is included in the output.

The second tag content of file is textually embedded having similar directive. The changed content is not included in the output. It is helpful when code from one jsp is required by several jsp’s.

**12)** **Explain JSP lifecycle methods?**

1) **jsplnit()**: The container calls this to initialize servlet instance. It is called only once for the servlet instance and preceded every other method.

2) **\_jspService()**: The container calls this for each request and passes it on to the objects.

3) **jspDestroy()**: It is called by the container just before destruction of the instance.

**13) How can information from one JSP be passed to another JSP?**

The tag <Jsp:param> allows us to pass information between multiple Jsp’s.

**14)** **Why does \_jspService() start with an ‘\_’ but other lifecycle methods do not?**

Whatever content made in a jsp page goes inside the \_jspService() method by the container. If it is override, the compiler gives an error, but the other 2 lifecycles can be easily override. So ‘\_’ shows that we cannot override this method.

**15)** **Explain the various scope values for <jsp:useBean> tag?**

<jsp:useBean> tag is used to use any java object in the jsp page. Some scope values are :

1)application

2)request

3)page

4)session

**16)** **What is Translation Phase in JSP?**

JSP engine translates and compiles a JSP file to a servlet. This servlet moves to the execution phase where requests and responses are handled. They are compiled for the first time they are accessed unless manually compiled ahead of time. The manual or explicit compilation is useful for long and convoluted programs.

**17)** **Perform a Browser Redirection from a JSP Page?**

<% response.sendRedirect(URL); %>

**18)** **How to forward a request to another source?**

<jsp:forward page="/Page2.jsp" />

**19) How can HTML Output be prevented from being cached?**

<% response.setHeader(“Cache-Control”, “no=store”);

response.setDateHeader(“Expires”, 0); %>

**20)** **Explain page Directives. Show attributes of page directives?**

Page Directives inform the JSP Engine about headers and facilities that the page receives from the environment. It is found at the top of all JSP pages. Its syntax is <%@ page attribute=”value”>

1)Session : It shows if a session data is available to the page.

2)Import : it shows packages that are imported.

3)isELIgnored : It shows whether EL expressions are ignored when JSP translates into a servlet.

4)contentType : it allows the user to specify the content type of page.

**21)** **What is a Servlet?**

A servlet is a Java technology and it is managed by a container called servlet engine. It generates dynamic content and interacts with client through Request and Response.

**22)** **What is a servlet context?**

Servlet context contains servlet view of Web application in which servlet will be running. By using the context,

* Log events
* Obtain URL references to resources
* Set and Store attributes

**23)** **What is life cycle of Servlet?**

Following is life cycle of Servlet:

* Loaded
* Initialized
* Destroy
* Unloaded

**24)** **What is called Session Tracking? Why session tracking is needed? What are the types of Session Tracking?**

Session tracking is used to maintain a state on the series of requests from the same user for a given period of time.

Every HTTP request needs to be captured by HTTP protocol and for that, state is captured. Tracking of state is called session tracking.

Types of session tracking:

* URL rewriting - URL rewriting is one of the methods of session tracking in which additional data is appended at the end of each URL. This additional data identifies the session.
* Hidden Form Fields
* Cookies - Cookies are used to store long term information that can be maintained without server interaction. Small and Medium size data are kept in a queue.
* Secure Socket Layer (SSL)

**25)** **What is the difference between ServletConfig and ServletContext?**

ServletConfig provides information about configuration of a servlet which is defined inside the web.xml file and it is a specific object for each servlet.

ServletContext is an application specific object and it is shared by all servlet. It belongs to one application in one JVM.

**26)** **What is Generic Servlet class?**

Generic servlet is the super class of all servlets. This class is extended by all other classes and it is protocol independent.

**27)** **What is HttpServlet and how it is different from GenericServlet?**

HttpServlet extends from GenericServlet and inherits the properties of Genericservlet. HttpServlet defines a HTTP protocol servlet while GenericServlet defines a generic, protocol-independent servlet.

**28) What are the uses of servlets?**

Servlets are used to process and store data submitted by HTML form, dynamic content, handle multiple request concurrently and manage state information on top of stateless HTTP.

**29) What are all the advantages of Servlet over CGI?**

Following are the advantages of Servlet over CGI:

* Cannot be run in an individual process.
* Servlet stays in the memory while requests. For every CGI request, you must load and start a CGI program.
* web.xml conveniences

**30) Difference between Java 7 and Java 8?**

A major difference between Java 7 and 8 is the introduction of Lambda Expressions in version 8. Lambda expressions are a way to simplify the verbosity behind the usage of `closures` in java.  
  
Closure is a way to define a function within the scope of another function. JavaScript is a good example of a language that inherently supports Closures.

|  |  |
| --- | --- |
| **Java7** | **Java8** |
| Strings in switch  Automatic resource management in try-statement  Improved type inference for generic instance creation, aka the diamond operator <>  Simplified varargs method declaration  Binary integer literals  Allowing underscores in numeric literals  Catching multiple exception types and rethrowing exceptions with improved type checking  Concurrency utilities under JSR 166  elliptic curve encryption | Lambda expressions  Improved Date Time API  Remove the Permanent Generation  Small VM  Parallel Array Sorting  Bulk Data Operations for Collections  Define a standard API for Base64 encoding and decoding  New Date & Time API  Provide stronger Password-Based-Encryption (PBE) algorithm implementations in the SunJCE provider |

**31) What is Reflection API in Java?**

- The Reflection API allows Java code to examine classes and objects at run time.  
- The new reflection classes allow you to call another class's methods dynamically at run time.  
- With the reflection classes, you can also examine an instance's fields and change the fields' contents.  
- It is also possible to instantiate new objects, invoke methods and get/set field values using reflection.  
- The Reflection API consists of the java.lang.Class class and the java.lang.reflect classes: Field, Method, Constructor, Array, and Modifier.

**32) Explain the significance of Classpath environment variables?**

- It is the list of directories which is used by JVM to find a class.  
- It tells the JVM and other applications related with the Java tools where to find the class libraries, including user-defined class libraries.  
- It is a set with 'SET' command.  
  
- **Syntax:** SET CLASSPATH=path1; path2...

- Using 'SET' command DOS prompt changes the current value of CLASSPATH.

**33) Can you explain about the access modifier in Java?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Access Modifiers** | **Same Class** | **Same Package** | **Subclass** | **Other packages** |
| public | Yes | Yes | Yes | Yes |
| protected | Yes | Yes | Yes | No |
| default | Yes | Yes | No | No |
| private | Yes | No | No | No |

**34)  Differences between String, StringBuffer and StringBuilder in Java?**

|  |  |  |
| --- | --- | --- |
| **String** | **StringBuffer** | **StringBuilder** |
| Immutable | Mutable | mutable |
| String operations such as append would be less efficient | String operations such as append would be more efficient, | String operations such as append would be more efficient |
| - | synchronized | Not synchronized. |
| - | versions 1.4 or below you’ll have to use StringBuffer. | StringBuilder was introduced in Java 1.5 |

**35) Can you explain about autoboxing and unboxing in Java?**

When primitive data types are automatically converted into it’s wrapper type, it is called boxing. The opposite operation of converting wrapper class objects to it's primitive type is known as unboxing.

ArrayList<Integer> list = new ArrayList<Integer>();  
list.add(1); //autoboxing - primitive to object  
int number = list.get(0); // unboxing

**36) Write a program that override equal() and hashcode()?**

|  |
| --- |
| class Person{  private String name;  private int age;   public Person(String name, int age){  this.name = name;  this.age = age;  }  public String getName() {  return name;  }  public void setName(String name) {  this.name = name;  }  public int getAge() {  return age;  }  public void setAge(int age) {  this.age = age;  }    public int hashCode(){  int hashcode = 0;  hashcode = age\*12;  hashcode += name.hashCode();  return hashcode;  }    public boolean equals(Object personObject){  if (personObject instanceof Person) {  Person pp = (Price) Person;  return (pp.name.equals(this.name) && pp.age == this.age);  } else {  return false;  }  } } |

**37) Explain the difference between RequestDispatcher and sendRedirect? How JSP handles runtime exceptions?**

- RequestDispatcher redirects the server-side by requesting and responding the objects. But sendRedirect redirects the Client-side with new request and responding the objects.  
- JSP handles runtime exceptions by using the errorPage attribute of the page directive.  
- By specifying ErrorPage = true the current page intended URL is redirected to the JSP.

**38) Explain the difference between ServletContext and PageContext. How to create request.getRequestDispatcher () and context.getRequestDispatcher()?**

- The ServletContext : Provides the information about the container.   
- PageContext : Provides the information about the Request.  
- To create request.getRequestDispatcher(path) we have to provide the relative path of the resource. But to create context.getRequestDispatcher (path) we have to provide the absolute path of the resource.