1. Byte code is a machine independent code.
2. Enum is Instance Variable, Constructors and methods.
3. LocalDateTime API makes it easy to work with date and time in java version 8.
4. Select the invalid main method syntax public static void main(String… args), public static void main(String args[]),public static void main(String []args),
5. getSession() is in the HttpServlet Class.
6. Static variable ki last updated value hi consider karta hai.
7. Final values can’t be changed.
8. List is
9. Hash Set’s basic functionalities are no repetition, unordered,not sorted (if we directly give the value then it won’t accept duplicates but when we pass objects then it will accept so Override the equals() and hashCode() and write down ur own logic.
10. For 9) Remember that equals compares the hash code and not the value so while passing the values directly, the jvm will check the values using pre-defined equals() and then if they are diff then it will assign it a hash code. But while passing objects, each object gets different hash code value so even if we have same values within the object it won’t be restricted that’s the reason we override the compareTo().
11. When we are creating an employee obj and adding it into the treeSet without implementing the comparables and without overriding the compareTo() then we will get the classCastException
12. If someone else want to use the same employee class but wants to compare the objects based on the names instead of the id which was used to compare in the comparables, then the new user can implement the class with the comparator and override the compare() and pass the comparator object as the parameter in the Collection class.
13. Illegal Assignments for byte, int, char and float

With explicit typecasting:

byte b=(byte) I;//Where i is an int

byte b=(short)10;

Without explicit typecasting:

float f=90.0f// Where 90.0 is a double and I converted it into float by adding an f at the end.

1. TreeMap me comparable ko implement karna padta hai sort karne ke liye.
2. InputStream and OutputStream classes are byte-oriented(for Reading binary type of files)(It can hold only 1 byte and then flush to read next byte), The reader and writer classes are character-oriented(for reading char files.)(It can hold many bytes at a time)
3. The superclass for java.sql.TimeStamp is java.util.date
4. Throw is the term used when exception is returned by the prg.
5. try{ return; } finally{system.out.println(“hello”);} output=hello
6. new obj().method(param)// this is a valid way of calling a method from the object obj.
7. If the called fn() throws an exception then that has to be handled by the calling fn().
8. The default priority level for thread is nom. (min=1,nom=5,max=10)
9. Thread t1=new Thread.join();Thread t2= new Thread.join(); in the main fn() means that the parent thread(Main fn) waits for all the child thread(t1,t2) to complete successfully and then the main fn() Starts. //This is because the .join() can control the Thread execution.
10. If a class extends an interface but the main() creates an obj of interface then compilation error at the line of creation of obj.
11. Comparator has the compare(), Using comparator we can create multiple sort sequence, and we need to modify the sort sequence for different prg.
12. Obj.flush() in serialization is used to force the file if it doesn’t reach the receiver.
13. In serialization while deserialization(ie reading or restoring data from file) the class that implements serializable uske parent ke constructor me jo kuch bhi hoga wo sab print hoga, and baki ke class ke jo constructors hai wo sab ignore ho jaega.
14. In serialization the child class’ constructor implicitly calls the super class’ constructor and goes on unless the super most class’ constructor is called.
15. If we are calling a procedure using a callable statement then we just need to use the execute() or executeQuery() though the procedure has a DML Query.
16. Logging can be set at runtime using a configuration file(we can use only API) ***\*Doubtful***
17. Class.forName();**DriverManager.registerDriver();** (Bold wala is optional and is used only when we create a new driver)
18. Default value for the data types. Int-0,Boolean-false
19. If we ignore the break; in switch then it will execute the next case as well
20. Byte, short, int, long, char, string, enum are accepted in switch.
21. Finalize() is called before Garbage Collection.
22. Final, public, private, protected, abstract, modifiers are used for classes and methods. And static for inner class and transient is extra for variables which are not serialized in the file.(transient is used to avoid sensitive data to persist in the table ie., it won’t get mapped in the table)
23. Object is created on heap.
24. String is immutable whereas String Buffer is mutable and thread safe(for memory pooling) whereas String builder is not thread safe.
25. Scanner ka delimiter hai space.
26. Abstract and exend can’t be used with final
27. Public static and final are by default for interface
28. ClassNotFound,SQLException,IOException,InterruptedException are checked Exceptions
29. ClassCastException,NotSerializable,NullPointer,Arithmatic,ArrayIndex,NumberFormat,inputTypeMismatch are unchecked ie Runtime Exception.
30. Base Class for all Exception is Throwable
31. CheckedExceptions are handled by extending Exception class, Unchecked Exceptions extend RuntimeException Class.
32. Table,view, index, sequence are DBMS Objects.
33. DQL-Select,DML-insert,update,delete, DCL-Grant,revoke, TCL-savepoint,rollback,commit
34. Execute() is present in Statement. ExecuteQuery is used to execute select query. ReturnType of executeQuery ResultSet(RecordType), ExecuteUpdate() se DML stmts.
35. Int is the return type for executeUpdate(). Boolean is the return type of execute(). IntegerArray is the return type of executeBatch();
36. AutoCommit is true by default in Java.
37. Collections is a class and Collection is an interface.
38. Difference between hash map is not sync and can contain null and hash table me ulta hai hash map.
39. @Suit in Junit is used to test all the functions within a class.
40. Methods in Runnable Interface, and Thread me Start(), Join,isAlive,SetPriority, GetPriority.

**Module-3**

1. Directive<%@ %>, Scripting<% %>, Action<jsp: action name> are the three main elements in the JSP.
2. Page, include and tag lib are the three types of directive.
3. Attributes of Page: import(a class1,class2,…), extends(to be derived from a class) but not required, errorPage, isErrorPage, isELIgnored(false by default), isThreadSafe(true by default), session(true by default), autoFlush(true by default), Buffer(8KB by default).
4. Attributes of taglib: URI, Prefix.
5. Attributes on include: page
6. Difference b/w include directive and include action is directive includes the jsp page during translation whereas the action includes it while displaying the result.
7. Forward is server side redirect. Is similar to <jsp:forward>
8. Include is used when I have a page which has a defined struct or design and I just have to include the response of some other page in the pre-defined sturct.
9. Life Cycle of JSP:\_ JSPinit(), \_JSPservice(),\_JSPdestroy() ->Translate to servlet-> init(), service(), destroy()
10. Phases of Servlet: Loading, Instantiation, Initialization(init()), Provide Serice(service()), Destroy(destroy).
11. Phases of JSP: Translation(to servlet), Compilation, Loading, Instantiation, Initialization(init()), Provide Serice(service()), Destroy(destroy).
12. Java.lang , javax.servlet, javax.servlet.http are the three packages imported by default in the JSP.
13. 9 Implicit objects in JSP: request(HttpServlet), response(HttpServletResponse), out(JSPWriter), config(ServletConfig), session(HttpSession), application(ServletContext), page(Object), pageContext(PageContext), exception(Throwable). //Brackets me return type hai in obj ka.
14. <jsp:forward><jsp:include><jsp:useBean id, class, scope, type><jsp:setProperty> <jsp:getProperty><jsp:param> ye sab actions hai JSP ke.
15. JSTL tags are used in the JSP to avoid Java code, and use tags for the same functionality.

**JPA:**

1. entityManagerFactory and entityManager

**Questions:**

1. Difference between ArrayList and LinkedList.
2. HashSet
3. Iterator
4. Abstract class and interface
5. Overloading and overriding
6. InstanceOf keyword
7. Final class and variable
8. Try with resource
9. Diff b/w enhanced for-loop and iterator.
10. Different Char Stream
11. File class.
12. DB Connectivity. Order is CallableStatement ->PreparedStatement->Statement(parent)
13. DML,DCL,TCL,DQL,DDL
14. Utility Functions.
15. Junit Annotations

**JSP:**

1. Functions of Servlet,ServletConfig, ServletContext, HttpServlet, HttpSession

**JPA:**

1. entityManagerFactory and entityManager
2. LifeCycle of JPA: merge, persist and detach.
3. @entity, @transient, @table-map entity with table @column,@id
4. Persistence.xml

**WEB SERVICE:**

1. Tags of WSDL file, SOAP, REST, jax.ws, jax.rs and their annotations.

**NOTE:** Mock API is used to create a dummy class to test the other class.

**UMA MAM**

1. Application sever supports Business(EJB) components.
2. Context ke saath if I use a requestDispatcher absolute address. So use slash.(unknownillegalstateException.)(jaha se diya wahi se consider karega.
3. Request -Relative address current state se batana padta hai.
4. SendRedirect changes the request so only session and context object scope use kar sakta hai.
5. SendRedirect changes the request so only session and context object scope use kar sakta hai.
6. Request.requestDispatcher can be used only within the application.
7. Response.sendRedirect can be used to redirect to servlet with some other application as well.
8. Request.getSession(),Request.getSession(true) gives u the old sessionId if exists else it will create a new sessionId, whereas Request.getSession(false) will just give if already exist else returns null.