Basic Programs asked by almost every company during Interview :  
1. Write a program to find factorial of the given number…  
2. Write a program to check whether the given number is even or odd.  
3. Write a program to swap two numbers using a temporary variable.  
4. Write a program to swap two numbers without using a temporary variable.  
5. Write a program to swap two numbers using bitwise operators.  
6. Write a program to find the greatest of three numbers.  
7. Write a program to find the greatest among ten numbers.  
8. Write a program to check whether the given number is a prime.  
9. Write a program to check whether the given number is a palindrome c number.  
10.Write a program to check whether the given string is a palindrome .  
11.Write a program to generate the Fibonacci series.  
12.Write a program to print”Hello World”without using semicolon anywhere in the code.  
13.Write a program to print a semicolon without using a semicolon anywhere in the code.  
14.Write a program to compare two strings without using strcmp() function.  
15.Write a program to concatenate e two strings without using strcat() function.  
16.Write a program to delete a specified line from a text file.  
17.Write a program to replace a specified line in a text file.  
18.Write a program to find the number of lines in a text file..  
19.Write a C program which asks the user for a number between 1 to 9 and shows the number. If the user  
inputs a number out of the specified range, the program should show an error and prompt the user for a  
valid input.  
20.Write a program to display the multiplication table of a given number..

Top of Form

[**Like**](https://www.facebook.com/groups/538351826334884/)Show More Reactions

[**Comment**](https://www.facebook.com/groups/538351826334884/)

Bottom of Form

[‪#‎Java\_Quick\_Reference‬](https://www.facebook.com/hashtag/java_quick_reference?source=feed_text) (Before attending any interview must go though all the questions.hope this helps !!) :

1. Java OOPS Concepts ?

Ans. Abstraction , Encapsulation , Inheritance and Polymorphism.

2. Difference between Abstract and Concrete Class ?

Ans. Abstract classes are only meant to be sub classed and not meant to be instantiated whereas concrete classes are meant to be instantiated.

3. Difference between Overloading and Overriding ?

Ans.

Overloading - Similar Signature but different definition , like function overloading.

Overriding - Overriding the Definition of base class in the derived class.

4. Types of Inner classes ?

Ans. Simple Inner Class, Local Inner Class, Anonymous Inner Class , Static Nested Inner Class.

5. Difference between TreeMap and HashMap ?

Ans. They are different the way they are stored in memory. TreeMap stores the Keys in order whereas HashMap stores the key value pairs randomly.

6. Difference between List , Sets and Maps ?

Ans. Lists - Members are stored in sequence in memory and can be accessed through index. Sets - There is no relevance of sequence and index. Sets doesn't contain duplicates whereas multiset can have duplicates. Maps - Contains Key , Value pairs.

7. Difference between Private , Public and Protected ?

Ans. Private - Not accessible outside object scope. Public - Accessible from anywhere. Protected - Accessible from anywhere within same package.

8. What is role of Synchronization in Java ?

Ans. Managing exclusive access of the code block to single thread at a time to maintain its integrity.

9. What is Default Access ?

Ans. Accessible only by the objects in the same package.

10. Difference between Vector and ArrayList ?

Ans. Vectors are synchronized whereas Array lists are not.

11. Difference between class and objects ?

Ans. Class is a template using which objects are created in memory.

12. Describe Garbage Collection in Java ?

Ans. Mechanism by which Java reclaims the memory inaccessible by the application.

13. What are the types of Polymorphism in Java ?

Ans. Static ( function overloading ) and Run time ( Virtual functions )

14. Different ways of using threads ?

Ans. Extending Thread class and implementing runnable interface.

15. Difference between Composition and Inheritence ?

Ans. Has a relationship vs Is a relationship.

16. What are Static variables ?

Ans. Correspond to class and not objects. All objects share that variable.

17. Difference between Interfaces and Abstract classes ?

Ans. Abstract classes have a body ( member elements ) where as Interfaces just have methods and static member declarations.

18. Explain Constructor ?

Ans. Its an operation that initializes the object.

19. What is Dynamic Binding ?

Ans. Association of function call to function definition during run time.

20. What is the use of Volatile Keyword ?

Ans. Volatile is a declaration that a variable can be accessed by multiple threads and hence shouldn't be cached.

21. Explain Serialization ?

Ans. Storing the state of an object in a file or other medium.

22. What is the use of Transient Keyword ?

Ans. keyword in Java is used to indicate that a field should not be serialized.

23. What are Static Methods ?

Ans. Can only operates on static variables.

25. What is a Final variable ?

Ans. Variable constant. Variable value can't be changed after instantiation.

26. What is a Final Method ?

Ans. Can't be overridden

27. What is a Final Class ?

Ans. Can't be sub classed.

28 What is an Immutable Object ?

Ans. Object that can't be changed after instantiation.

29. Difference between checked and unchecked exceptions ?

Ans. For checked exceptions compiler throws a errors if they are not checked whereas unchecked exceptions and caught during runtime only and hence can't be checked.

30 Give an Example of checked and unchecked exception ?

Ans. ClassNotFoundException is checked exception whereas NoClassDefFoundError is a unchecked exception.

31. What is a Singleton Class ?

Ans. Using which only 1 object can be created.

32. Name few Java exceptions ?

Ans. IndexOutofBound , NoClassDefFound , OutOfMemory , IllegalArgument.

33. Name few Design Patterns used for designing Java applications ?

Ans. Singleton , Factory , Abstract Factory , Proxy , Command , Builder.

34. Give an example of Runtime Polymorphism ?

Ans. Action Mapping in a web application , Generics.

35. Which of the following is tightly bound ? Inheritance or Composition ?

Ans. Inheritence.

36. Give an Example of Generics in Java 5 ?

Ans. List<number> aList = new ArrayList <number>

37. How can we make sure that a code segment gets executed even in case of uncatched exceptions ?

Ans. By putting it within finally.

38. Can a class extend an interface ?

Ans. No Class implement the interface. An interface can extend another interface.

39. Does Java support multiple inheritence ?

Ans. No. We can't extend multiple classes but can implement multiple interfaces.

40. Can we use an interface reference for referring to an object ?

Ans Yes we can do it for object implementing that interface.

41. What do you mean by return type Void ?

Ans. Returns Nothing.

42. Is Java Compiled or an Interpreted language ?

Ans Both

43. What is a class file ?

Ans. Compiled Java files.

44. Why Java is a called a platform independent language ?

Ans. Java creates the pre compiled files ( class files ) which can then be interpreted on multiple platforms.

45. Explain the use of "Native" keyword ?

Ans. Used in method declarations to specify that the method is not implemented in the same Java source file, but rather in another language

46. What is "super" used for ?

Ans. Used to access members of the base class.

47. What is "this" keyword used for ?

Ans. Used to represent an instance of the class in which it appears.

48. What is the default value of local variable ?

Ans. The local variables are not initialized to any default value.

49. What will happen if we declare a class abstract as well as final ?

Ans. Compile time error.

50 Encapsulation ?

Ans. Encapsulation is the mechanism that binds together code and data it manipulates and keeps both safe from outside interference and misuse

51. Explain Inheritence ?

Ans. Inheritance is the process by which one object acquires the properties of another object.

52. Explain Polymorphism ?

Ans. Polymorphism is the feature that allows one interface to be used for general class actions

53. Difference between assignment and initialization ?

Ans. Assignment can be done anytime but initialization can be done only once during object instantiation.

54. Difference between boolean and Boolean ?

Ans. boolean is a primitive type whereas Boolean is a class.

55. What is an Immutable object ?

Ans. Objects that can't be changed after initialization.

56. Give an example of immutable class ?

Ans. String

57. What is Casting ?

Ans. Process to convert one data type to another.

58. Explain Finalize () ?

Ans. finalize() method is used just before an object is destroyed.

59. What are Marker Interfaces ?

Ans. These are the interfaces which have no declared methods.

60. Name few Java marker interfaces ?

Ans. Serializable and cloneable.

61. Is runnable a Marker interface ?

Ans. No , it has run method declared.

62. Difference between String and StringBuffer ?

Ans. String is an immutable class where StringBuffer is not.

63. Difference between Process and Thread ?

Ans. Process is a program in execution whereas thread is a separate path of execution in a program.

64. Explain Thread states ?

Ans. ready, running, waiting and dead.

65. What is a Deadlock ?

Ans. When two threads are waiting each other and can’t precede the program is said to be deadlock.

66. Difference between Serialization and Deserialization ?

Ans. Serialization is the process of writing the state of an object to a byte stream. Deserialization is the process of restoring these objects.

67. Difference between Java 1.4 and Java 5 ?

Ans. Generics , Autoboxing , Enum and Static Imports.

68. Explain Autoboxing ?

Ans. Autoboxing is the automatic conversion that the Java compiler makes between the primitive types and their corresponding object wrapper classes

69. What is an Enum type ?

Ans. An enum type is a special data type that enables for a variable to be a set of predefined constants

70. What are Wrapper Classes ?

Ans. A wrapper class is any class which "wraps" or "encapsulates" the functionality of another class or component.

71. What are Primitive Wrapper Classes ?

Ans. A Wrapper Class that wraps or encapsulates the primitive data type.

72. What is automatic conversion of primitive data type to its Wrapper class called ?

Ans. Autoboxing.

73. What Design pattern Wrapper Classes implement ?

Ans. Adapter.

74. What is "Import" used for ?

Ans. Enables the programmer to abbreviate the names of classes defined in a package.

75. Explain Servlet Chaining ?

Ans. Servlet chaining is a technique in which two or more servlets can cooperate in servicing a single request.

76. What is the difference between set and list?

Ans. Set stores elements in an unordered way but does not contain duplicate elements, whereas list stores elements in an ordered way but may contain duplicate elements. List can also holds null elements while Set can't.

77. Difference between HashMap and HashTable ?

Ans. The HashMap class is roughly equivalent to Hashtable, except that it is unsynchronized and permits nulls.

78. What is an Iterator ?

Ans. Some of the collection classes provide traversal of their contents via a java.util.Iterator interface. This interface allows you to walk through a collection of objects, operating on each object in turn.

79. What is JVM ?

Ans. Java Virtual Machine or JVM is an abstract which provides the runtime environment for the Java byte code to run.

80. What is JDK ?

Ans. Java Development Kit or JDK is physical entity that comprises of JRE and Development Tools.

81. What is JRE ?

Ans. Java Run Time Environment is the implementation of JVM.

82. Different types of memory used by JVM ?

Ans. Class , Heap , Stack , Register , Native Method Stack.

83. What is a Class Loader ?

Ans. Part of JVM which is used to load classes and interfaces.

84. Does Static Public Void instead of Public static Void gives compilation error ?

Ans. No

85. Does constructor returns any value ?

Ans. Yes , the current instance or object of the class.

86. Can we make constructor final ?

Ans. No

87. Why main method is static ?

Ans. Object needs to be created for calling non static methods.

88. What is a Static Block ?

Ans. It is executed before main method at the time of class loading.

89. Can we execute program without mail method ?

Ans. Yes , one way is through static block.

90. What is Cloning ?

Ans. Keyword to create the exact copy of the object.

91. Can we overload main method ?

Ans. Yes.

92. Can we declare main method as final ?

Ans. Yes.

93. What is Static binding ?

Ans. Associating function call and its definition during compile time.

94. What is Dynamic Binding ?

Ans. Associating function call and its definition during runtime.

95. Can we declare an interface method static ?

Ans. No

96. Can an interface be final ?

Ans. No

97. Can we declare interface methods as private ?

Ans. No, they are implicitely public.

98. What is a Static import ?

Ans. By static import , we can access the static members of a class directly without prefixing it with the class name.

99. Base class for Error and Exceptions ?

Ans. Throwable

100. Difference between StringBuffer and StringBuilder ?

Ans. StringBuffer is synchronized whereas String Builder is not.

101. Default Package for Collection classes ?

Ans. Java.Util

102. Difference between List and Queue ?

Ans. In queue access can only happen at the ends whereas we can access any element of the list.

103. Difference between Vector and ArrayList ?

Ans. Vectors are synchronized whereas ArrayList are not.

104. Which is the base interface for all collection classes ?

Ans. Collection interface.

105. How can we make HashMap synchronized ?

Ans. Map m = Collections.synchronizedMap(hashMap);

106. Difference between Map and HashMap ?

Ans. Map is an interface where HashMap is the concrete class.

107. Property Class ?

Ans. The properties class is a subclass of Hashtable that can be read from or written to a stream.

108 Why Java is a preferred technology for Web Applications ?

Ans. Web application involves dealing with multiple requests and thread and hence deals with lot of memory allocation and de allocation. Hence Java with its intrinsic capability of Garbage collection offer a perfect platform.

109. What are concepts introduced with Java 5 ?

Ans. Generics , Enums , Autoboxing , Annotations and Static Import.

110. Difference between get and post ?

Ans. Request parameters in get are passed within url whereas with post these are passed within post body as object.

111. Difference between C++ and Java ?

Ans. We have pointers in C++ , We have garbage collection in Java.

112. Name few Design Patterns ?

Ans. Singleton , Adapter , Factory , Abstract Factory , Builder , Facade , Observer , Proxy , Prototype.

113. Give an Example of Factory Design Pattern ?

Ans. Action Mapping within Web Application.

114. Give an example of Adapter design pattern ?

Ans. Wrapper Classes.

115. Can we override static methods ?

Ans. No

116. What precaution should be used while dealing with static variables ?

Ans. Static variables are shared by all objects of the class and hence each instance may update it with its local value.

117. Can we override main method ?

Ans. No

118. Difference between server side and client side validation ?

Ans. Client side validation is done using scripting technologies like javascript before the form is submitted to server whereas server side validation is done after the form is submitted to server.

119. Difference between sleep and wait ?

Ans. Sleep puts thread aside for exactly the time specified. Wait causes a wait of up to time specified. A thread could stop waiting earlier if it receives the notify() or notifyAll() call.

120. Name Wrapper classes available for primitive types ?

Ans.

boolean - java.lang.Boolean  
byte - java.lang.Byte  
char - java.lang.Character  
double - java.lang.Double  
float - java.lang.Float  
int - java.lang.Integer  
long - java.lang.Long  
short - java.lang.Short  
void - java.lang.Void

121. Explain System.out.println ?

Ans. System is the class having static variable out that defaults to system console and println is the overloaded method.

122. Explain public static void main() ?

Ans. public means it has public access, static means that the method can be invoked without instantiating the object, void means it returns nothing.

123. What is Java Byte Code ?

Ans. Java bytecode is the form of instructions that the Java virtual machine executes.

124. What is the relation between class file and byte code ?

Ans. Class files holds the instructions in byte code.

125. How can we make the object immutable ?

Ans. By making it a final class and making all members as private with those being initialized through constructor.

126. What is the purpose of making an object immutable ?

Ans. Since they cannot change state, they cannot be corrupted by thread interference or observed in an inconsistent state.

127. Explain the scenerios to choose between String , StringBuilder and StringBuffer ?

Ans.   
If the Object value will not change in a scenario use String Class because a String object is immutable.  
If the Object value can change and will only be modified from a single thread, use a StringBuilder because StringBuilder is unsynchronized(means faster).  
If the Object value may change, and can be modified by multiple threads, use a StringBuffer because StringBuffer is thread safe(synchronized).  
128. Explain java.lang.OutOfMemoryError ?

Ans. This Error is thrown when the Java Virtual Machine cannot allocate an object because it is out of memory, and no more memory could be made available by the garbage collector.

129. Difference between arraylist and linkedlist ?

Ans. Array list maintains indices like array whereas Linked List maintain pointers to the next element.

130. Difference between hashtable and hashmap ?

Ans. Hashtable are synchronized whereas hashmaps are not. Hashtable doesn't allow null values whereas HashMaps allows null values.

131. Can we have multiple servlets in a web application and How can we do that ?

Ans. Yes by making entries in web.xml

132. Are static members serialized ?

Ans. Static members don't belong to a object and hence are not serialized.

133. How can we manage Error Messages in the web application ?

Ans. Within message.properties file.

134. What is the use of Marker interface if they don't hold anything ?

Ans. As the name suggest , They just mark a class and just indicate something to compiler and JVM.

135. Is JVM, a compiler or interpretor ?

Ans. Its an interpretor.

136. Is Synchronization an overhead ?

Ans. Synchronization has relevance only if a code segment can be accessed by multiple threads and hence may result in ambiguity. Yes Its an overhead if simultaneous access doesn't pose any problem.

137. Difference between implicit and explicit type casting ?

Ans. An explicit conversion is where you use some syntax to tell the program to do a conversion whereas in case of implicit type casting you need not provide the data type.

138. Can we have an abstract class without abstract methods ?

Ans. Yes

139. Can we have abstract methods in a class which is not Abstract ?

Ans. No

140. Are list , Map and Set Classes or Interfaces ?

Ans. These are interfaces.

141. Difference between .equals and == ?

Ans. .equals is the method of object class and is used to match the content of object whereas == matches the objects itself. So == looks to see if they are same object.

142. What are the methods of Object class ?

Ans. toString , clone , finalize , hashcode , equals.

143. What are the class loaders ?

Ans. JVM loads the classes through class loaders.

144. What are the different class loaders used by JVM ?

Ans. Bootstrap , Extension and System.

145. difference between loadClass and Class.forName ?

Ans. loadClass only loads the class but doesn't initialize the object whereas Class.forName initialize the object after loading it.

146. How do we load the DB drivers ?

Ans. Using Class.forName

147. Have you ever destroyed objects yourself ?

Ans. Java by its mechanism of Garbage collection does this itself.

148. Should we override finalize method ?

Ans. Finalize is used by Java for Garbage collection. It should not be done as we should leave the Garbage Collection to Java itself.

149. Will finally be executed for handled or unhandled exceptions ?

Ans. Both.

150. Difference between protected and default Modifier ?

Ans. With protected member is accessible in subclass whereas with Default Member is not accessible in subclass.

151. Can we overload methods on a single parameter with data types as int and long. For example - int calculate(int x) and int calculate(long x) ?

Ans. Yes

152. What will happen if we call such method using a number lets say calculate(5) ?

Ans. It will call first method with int argument.

153. Then whats the use of second method. Will that ever be called ?

Ans. we can call by having argument as 5L.

154. What is servlet Chaining ?

Ans. Multiple servlets serving the request in chain.

155. In the following code

String str1 = new String("abc");  
String str2 = new String("abc");

Which of the following will be true.

str == str2  
str.equal(str2)

Ans. str.equal(str2) will be true but str==str2 will be false

156. In the following code

String str1 = new String("abc");  
String str2 = new String("xyz");  
str1=str2;

Which of the following will be true.

str == str2  
str.equal(str2)

Ans. both str.equal(str2) and str==str2 will be true

157. What is keyword instanceOf used for ?

Ans. It is used to see if an object is an instance of a specific class.

158. If we obj is an object of Class A and Class A extends Class B , What will obj instanceof B will return ?

Ans. True

159. What is assert keyword used for ?

Ans. The assert keyword is used to make an assertion—a statement which the programmer believes is always true at that point in the program. This keyword is intended to aid in testing and debugging.

160. Any example of Serialization in your application ?

Ans. Properties Class.

161. Difference between Factory and Abstract Factory Design Pattern ?

Ans. Factory Pattern deals with creation of objects delegated to a separate factory class whereas Abstract Factory patterns works around a super-factory which creates other factories.

162. Difference between Factory and Builder Design Pattern ?

Ans. Builder pattern is the extension of Factory pattern wherein the Builder class builds a complex object in multiple steps.

163. Difference between Proxy and Adapter ?

Ans. Adapter object has a different input than the real subject whereas Proxy object has the same input as the real subject. Proxy object is such that it should be placed as it is in place of the real subject.

164. Difference between Adapter and Facade ?

Ans. The Difference between these patterns in only the intent. Adapter is used because the objects in current form cannot communicate where as in Facade , though the objects can communicate , A Facade object is placed between the client and subject to simplify the interface.

165. Difference between Builder and Composite ?

Ans. Builder is a creational Design Pattern whereas Composite is a structural design pattern. Composite creates Parent - Child relations between your objects while Builder is used to create group of objects of predefined types.

166. Example of Chain of Responsibility Design Pattern ?

Ans. Exception Handling Throw mechanism.

167. Example of Observer Design Pattern ?

Ans. Listeners.

168. Difference between Factory and Strategy Design Pattern ?

Ans. Factory is a creational design pattern whereas Strategy is behavioral design pattern. Factory revolves around the creation of object at runtime whereas Strategy or Policy revolves around the decision at runtime.

169. Shall we use abstract classes or Interfaces in Policy / Strategy Design Pattern ?

Ans. Strategy deals only with decision making at runtime so Interfaces should be used.

170. Which kind of memory is used for storing object member variables and function local variables ?

Ans. Local variables are stored in stack whereas object variables are stored in heap.

171. Why do member variables have default values whereas local variables don't have any default value ?

Ans. member variable are loaded into heap, so they are initialized with default values when an instance of a class is created. In case of local variables, they are stored in stack until they are being used.

172. What is the default value for Object Reference in Java ?

Ans. Null

173. What is a Default Constructor ?

Ans. The no argument constructor provided by Java Compiler if no constructor is specified.

174. Will Compiler creates a default no argument constructor if we specify only multi argument constructor ?

Ans. No, Compiler will create default constructor only if we don't specify any constructor.

175. Can we overload constructors ?

Ans. Yes.

176. What will happen if we make the constructor private ?

Ans. We can't create the objects directly by invoking new operator.

177. How can we create objects if we make the constructor private ?

Ans. We can do so through a static public member method.

178. Is that the only way to initialize the object in that case ?

Ans. We can also do so using static block.

179. Is constructor inherited ?

Ans. No

180. What will happen if we remove the static keyword from main method ?

Ans. Program will compile but will give a "NoSuchMethodError" during runtime.

181. Which is the super class of all classes in java ?

Ans. Object Class

182. Why Java don't use pointers ?

Ans. Pointers are vulnerable and slight carelessness in their use may result in memory problems and hence Java intrinsically manage their use.

183. Can we use both "this" and "super" in a constructor ?

Ans. No, because both this and super should be the first statement.

184. Can we achieve runtime polymorphism by data members ?

Ans. No

185. Do we need to import java.lang.package ?

Ans. No, It is loaded by default by the JVM.

186. Can i import same package / class twice ?

Ans. Compiler doesn't give any error if we try to import same package or class twice but internally JVM will load the class only once.

187. Is it necessary that each try block to be followed by catch block ?

Ans. It should be followed by either catch or finally block.

188. Can finally block be used without catch ?

Ans. Yes but should follow "try" block then.

189. Can an exception be rethrown ?

Ans. Yes

190. What is exception propogation ?

Ans. Passing the exception object to the calling method.

191. What is the use of toString method ?

Ans. It provides the string representation of an object.

192. Difference between nested and inner classes ?

Ans. Inner classes are non static nested classes.

193. What is a nested interface ?

Ans. Any interface declared inside a class or an interface. It is static by default.

194. Can an unreferenced object be referenced again ?

Ans. Yes.

195. What kind of thread is garbage collector thread ?

Ans. Dameon Thread.

196. What is an Externalizable interface ?

Ans. Externalizable interface is used to write the state of an object into a byte stream in compressed format.

197. Difference between serializable and externalizable interface ?

Ans. Serializable is a marker interface whereas externalizable is not.

198. What is reflection ?

Ans. It is the process of examining / modifying the runtime behaviour of an object at runtime.

199. What is the purpose of System Class ?

Ans. To provide access to the system resources.

200. What is Locale ?

Ans. A Locale object represents a specific geographical, political, or cultural region.

201. Situations where "this" keyword can be used ?

Ans.

a. While accessing the member variable having same name as local variable.  
b. constructor chaining  
c. pass the current object as a parameter to another method.

202. Can we declare interface method's private ?

Ans. No

203. What is the problem with following code ?

if(x == null && x.getMethod() == true)

Ans. It will throw NPE in case x is null.

204. Can we instantiate the object of derived class if parent constructor is protected ?

Ans. Yes.

206. Can we declare an abstract method private ?

Ans. No Abstract methods can only be declared protected or public.

207. Can Abstract methods in parent class have a body ?

Ans. No.

208. Can we declare a class static ?

Ans. We cab only declare the inner class as static but not the top level class.

209. Can we reduce the visibility of the overridden method ?

Ans. No.

210. Can we make an object final i.e immutable using final keyword ?

Ans. No, only object reference can be made final. This means that the reference cannot be reassigned.

211. What are the design considerations while making a choice between using interface and abstract class ?

Ans. Keep it as a Abstract Class if its a "Is a" Relationsship and should do subset/all of the functionality. Keep it as Interface if its a "Should Do" relationship.

212. What are new features introduced with Java 8 ?

Ans. Lambda Expressions , Interface Default and Static Methods , Method Reference , Parameters Name , Optional , Streams, Concurrency.

Q213. What is a Lambda Expression ? What's its use ?

Ans. Its an anonymous method without any declaration. Lambda Expression are useful to write shorthand Code and hence saves the effort of writing lengthy Code. It promotes Developer productivity, Better Readable and Reliable code.

Q214. What are Default Methods ?

Ans. With Java 8, We can provide method definitions in the Interfaces that gets carried down the classes implementing that interface in case they are not overridden by the Class. Keyword "default" is used to mark the default method.

Q215. Can we have a default method definition in the interface without specifying the keyword "default" ?

Ans. No. Compiler complains that its an abstract method and hence shouldn't have the body.

Q216. Can we use static method definitions in Interfaces ?

Ans. Yes, Effective Java 8.

Q217. Can we access Interface static method using Interface references ?

Ans. No, only using Interface Name.

Q218. What is the difference between Data Type and Data Structure ?

Ans. Data type: a set of values together with operations on that type. Data structure: a physical implementation of a data type

Q219. Is it correct to say that Interfaces are abstract data types ?

Ans. No. Data Type holds data whereas Interface doesn't hold anything. Interface is a contract about how to communicate with the underlying Class.

Q220. Which sorting algorithm is used by Collections.sort() in Java ?

Ans. The sorting algorithm is a modified mergesort. This algorithm offers guaranteed n log(n) performance.