**Subjective Questions**

**1.Are JVM's platform independent?**

JVM's are not platform independent. JVM's are platform specific run time implementation provided by the vendor.

**2.What is the difference between creating String as new() and literal?**

When we create string with new() Operator, it’s created in heap and not added into string pool while String created using literal are created in String pool itself which exists in PermGen area of heap.

String s = new String("Test");  
   
does not  put the object in String pool , we need to call String.intern() method which is used to put  them into String pool explicitly. its only when you create String object as String literal e.g. String s = "Test" Java automatically put that into String pool

**3.What will happen if you call return statement or System.exit on try or catch block ? will finally block execute?**

This is a very *popular tricky Java question* and its tricky because many programmer think that finally block always executed. This question challenge that concept by putting return statement in try or catch block or calling System.exit from try or catch block. Answer of this tricky question in Java is that finally block will execute even if you put return statement in try block or catch block but finally block won't run if you call System.exit form try or catch.

**4.  If a method throws NullPointerException in super class, can we override it with a method which throws RuntimeException?**

One more tricky Java questions from overloading and overriding concept. Answer is you can very well throw super class of RuntimeException in overridden method but you can not do same if its checked Exception.

**5. What is the difference between ArrayList and LinkedList ?**

Please pay special attention as this is probably one of the most widely asked interview questions.

We aren’t going to state the properties of each in this question. What we are looking for are the differences. The prime areas where the two stand apart are as follows :

|  |  |
| --- | --- |
| **Arraylist** | **Linklist** |
| Random access. | Sequential access.The control traverses from the first node to reach the indexed node. |
| Only objects can be added. | The LinkedList is implemented using nodes linked to each other. Each node contains a previous node link, next node link, and value, which contains the actual data |

**Or**

**5.  What differences exist between HashMap and Hashtable ?** Both the [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html) and [Hashtable](http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html) classes implement the Map interface and thus, have very similar characteristics. However, they differ in the following features:

* A [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html) allows the existence of null keys and values, while a [Hashtable](http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html) doesn’t allow neither null keys, nor null values.
* A [Hashtable](http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html) is synchronized, while a [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html) is not. Thus, [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html) is preferred in single-threaded environments, while a [Hashtable](http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html) is suitable for multi-threaded environments.
* A [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html) provides its set of keys and a Java application can iterate over them. Thus, a [HashMap](http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html) is fail-fast. On the other hand, a [Hashtable](http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html) provides an [Enumeration](http://docs.oracle.com/javase/7/docs/api/java/util/Enumeration.html) of its keys.
* The [Hashtable](http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html) class is considered to be a legacy class.

**6.** **What does Connection pooling mean ?** The interaction with a database can be costly, regarding the opening and closing of database connections. Especially, when the number of database clients increases, this cost is very high and a large number of resources is consumed.A pool of database connections is obtained at start up by the application server and is maintained in a pool. A request for a connection is served by a [connection residing in the pool](http://examples.javacodegeeks.com/enterprise-java/hibernate/hibernate-connection-pool-configuration-with-c3p0-example/). In the end of the connection, the request is returned to the pool and can be used to satisfy future requests.

7. **Can I declare a data type inside loop in java?**

* Any Data type declaration should not be inside the loop. Its possible but not recommended.

**8. Is it possible to declare abstract method with default and with protected modifier?**

* Yes we can declare abstract methods with default.

1. package Abstraction;
2. public abstract class AbstractClassExample {
4. abstract void add();
6. void show(){ // normal method
7. System.out.println("this is concrete method present in abstract class");
8. }
9. }

* Yes we can declare abstract methods as protected.

1. package Abstraction;
2. public abstract class AbstractClassExample {
4. protected abstract void add();
6. void show(){ // normal method
7. System.out.println("this is concrete method present in abstract class");
8. }
9. }

**9.**

|  |
| --- |
| void start() {  A a = new A();  B b = new B();  a.s(b);  b = null; /\* Line 5 \*/  a = null; /\* Line 6 \*/  System.out.println("start completed"); /\* Line 7 \*/  }  When is the B object, created in line 3, eligible for garbage collection? |
| |  |  | | --- | --- | | [**A.**](javascript:%20void%200;) | after line 5 | | [**B.**](javascript:%20void%200;) | after line 6 | | [**C.**](javascript:%20void%200;) | after line 7 | | [**D.**](javascript:%20void%200;) | There is no way to be absolutely certain. |   **Answer:** Option **D** |

**10.**

|  |
| --- |
| What will be the output of the program?  class PassS  {  public static void main(String [] args)  {  PassS p = new PassS();  p.start();  }  void start()  {  String s1 = "slip";  String s2 = fix(s1);  System.out.println(s1 + " " + s2);  }  String fix(String s1)  {  s1 = s1 + "stream";  System.out.print(s1 + " ");  return "stream";  }  } |
| |  |  | | --- | --- | | [**A.**](javascript:%20void%200;) | slip stream | | [**B.**](javascript:%20void%200;) | slipstream stream | | [**C.**](javascript:%20void%200;) | stream slip stream | | [**D.**](javascript:%20void%200;) | slipstream slip stream |   **Answer:** Option **D** |

**11.**

|  |
| --- |
| public class Outer  {  public void someOuterMethod()  {  //Line 5  }  public class Inner { }    public static void main(String[] argv)  {  Outer ot = new Outer();  //Line 10  }  }  Which of the following code fragments inserted, will allow to compile? |
| |  |  | | --- | --- | | [**A.**](javascript:%20void%200;) | new Inner(); //At line 5 | | [**B.**](javascript:%20void%200;) | new Inner(); //At line 10 | | [**C.**](javascript:%20void%200;) | new ot.Inner(); //At line 10 | | [**D.**](javascript:%20void%200;) | new Outer.Inner(); //At line 10 |   **Answer:** Option **A**  **Explanation:**  Option A compiles without problem.  Option B gives error - non-static variable cannot be referenced from a static context.  Option C package ot does not exist.  Option D gives error - non-static variable cannot be referenced from a static context. |

**12.** **What is the purpose of Void class?**

The Void class is an uninstantiable placeholder class to hold a reference to the Class object representing the primitive Java type void.

**13. What is JIT and its use?**

Really, just a very fast compiler… In this incarnation, pretty much a one-pass compiler — no offline computations. So you can’t look at the whole method, rank the expressions according to which ones are re-used the most, and then generate code. In theory terms, it’s an on-line problem

**14.** **Difference between static and dynamic class loading.**

Static class loading: The process of loading a class using new operator is called static class loading. Dynamic class loading: The process of loading a class at runtime is called dynamic class loading.  
Dynamic class loading can be done by using Class.forName(….).newInstance().

**15.** **What is the purpose of apache tomcat or difference between apache and tomcat?**

Apache server is a standalone server that is used to test servlets and create JSP pages. It is free and open source that is integrated in the Apache web server. It is fast, reliable server to configure the applications but it is hard to install. It is a servlet container that includes tools to configure and manage the server to run the applications. It can also be configured by editing XML configuration files.

**16.** **How many objects are created in the following piece of code?**

MyClass c1, c2, c3;  
c1 = new MyClass ();  
c3 = new MyClass ();  
Answer: Only 2 objects are created, c1 and c3. The reference c2 is only declared and not initialized.

**17.** class IncDec {

public **static** **void** main(String args[]) {

**int** a = 1;

**int** b = 2;

**int** c;

**int** d;

c = ++b;

d = a++;

c++;

System.out.println("a = " + a);

System.out.println("b = " + b);

System.out.println("c = " + c);

System.out.println("d = " + d);

}

}

## Output :

a = 2

b = 3

c = 4

d = 1

**18. What will this return 3\*0.1 == 0.3? true or false?**(answer)  
This is one of the really tricky questions. Out of 100, only 5 developers answered this question and only of them have explained the concept correctly. The short answer is false because some floating point numbers can not be represented exactly.

19. **What is a.hashCode() used for? How is it related to a.equals(b)?**([answer](http://javarevisited.blogspot.sg/2011/10/override-hashcode-in-java-example.html))  
hashCode() method returns an int hash value corresponding to an object. It's used in hash based collection classes e.g Hashtable, HashMap, LinkedHashMap and so on. It's very tightly related to equals() method. According to Java specification, two objects which are equal to each other using equals() method must have same hash code.

20. **Can I write my own container class and use it in the for-each loop?**  
Yes, you can write your own container class. You need to implement the Iterable interface if you want to loop over advanced for loop in Java, though. If you implement Collection then you by default get that property.

**Objective Questions**

**Q1: class Test{**

**public static void main(String args[]){**

**int String =10;**

**int Runnable=20;**

**System.out.println("Sum of two numbers: "+String+Runnable);**

**}**

**}**

**What will be the output after running this java class??**

**Option 1: At runtime: Exception in thread "main" java.lang.ClassCastException: java.lang.Integer cannot be cast to java.lang.String**

**Option 2: Compile time Error**

**Option 3: Sum of two numbers: 30**

**Option 4: None of them.**

**Correct output is : Option 3:**

**Explanation: String and Runnable are not reserved words, we can use it but according to coding standards its not advisable.**

**Q2: class Test{**

**public static int main(String args[]){**

**int flag=0;**

**int number=40;**

**if(number%2==0)**

**flag=1;**

**else**

**flag=0;**

**if(flag)**

**System.out.println("no. is even");**

**else**

**System.out.println("no. is odd");**

**}**

**}**

**What will be the output after running this java class??**

**Option 1: error: incompatible types if(flag) ^ required: boolean found: int**

**Option 2: no. is even**

**Option 3: no. is odd**

**Option 4: None of them.**

**Explanation:**

**Option 1 is the correct answer because java only accepts Boolean value in condition statements like if, while etc , whereas in c, c++ u can also pass Boolean as well as integer value that is 0 and 1.**

**Q3:  Under what conditions is an object's finalize() method invoked by the garbage collector?**

**Option 1: When it detects that the object has become unreachable**

**Option 2:** [**As soon as object is set as null.**](javascript:void(0);)

**Option 3: At fixed intervalm it checks for null value.**

**Option 4: None of them.**

**Explanation: option 1: The garbage collector invokes an object's finalize() method when it detects that the object has become unreachable.**

**Q4: When finally block gets executed?**

**Option 1:** [**Always when try block get executed, no matter exception occured or not.**](javascript:void(0);)

**Option 2:** [**Always when a method get executed, no matter exception occured or not.**](javascript:void(0);)

**Option 3:** [**Always when a try block get executed, if exception do not occur.**](javascript:void(0);)

**Option 4:** [**Only when exception occurs in try block code.**](javascript:void(0);)

**Explanation: option 1: Always when try block get executed, no matter exception occured or not.**

**Q5 - Which of the following stands true about default modifier of class members?**

[**A - By default, variables, methods and constructors can be accessed by subclass only.**](javascript:void(0);)

[**B - By default, variables, methods and constructors can be accessed by any class lying in any package.**](javascript:void(0);)

[**C - By default, variables, methods and constructors can be accessed by any class lying in the same package.**](javascript:void(0);)

[**D - None of the above.**](javascript:void(0);)

**Answer : C**

**Explaination**

**By default, variables, methods and constructors can be accessed by any class lying in the same package.**

**Q6 - What is currentThread()?**

[**A - It is a Thread public static method used to obtain a reference to the current thread.**](javascript:void(0);)

[**B - It is a thread's instance method used to get thread count.**](javascript:void(0);)

[**C - It is a object's public static method used obtain a reference to the current thread.**](javascript:void(0);)

[**D - It is a object's instance method used to get thread count.**](javascript:void(0);)

**Answer : A**

**Explaination**

**currentThread() is a public static method of Thread class used to obtain a reference to the current thread.**

**Q7 - What is JRE?**

[**A - JRE is a java based GUI application.**](javascript:void(0);)

[**B - JRE is an application development framework.**](javascript:void(0);)

**C - JRE is an implementation of the Java Virtual Machine which executes Java programs.**

[**D - None of the above.**](javascript:void(0);)

**Answer : C**

**Explaination**

**Java Runtime Environment is an implementation of the Java Virtual Machine which executes Java programs. It provides the minimum requirements for executing a Java application**

**Q8 - Which of the following is Faster, StringBuilder or StringBuffer?**

**A - StringBuilder**

[**B - StringBuffer**](javascript:void(0);)

[**C - Both of the above.**](javascript:void(0);)

[**D - none of the above.**](javascript:void(0);)

**Answer : A**

**Explaination**

**StringBuilder is faster than StringBuffer.**

**Q9 - In which case, a program is not expected to recover?**

**A - If an error occurs.**

[**B - If an exception occurs.**](javascript:void(0);)

[**C - Both of the above.**](javascript:void(0);)

[**D - None of the above.**](javascript:void(0);)

**Answer : A**

**Explaination**

**A program is not expected to recover if an error occurs.**

**Q10 - Which of the following is true about super class?**

[**A - Variables, methods and constructors which are declared private can be accessed only by the members of the super class.**](javascript:void(0);)

[**B - Variables, methods and constructors which are declared protected can be accessed by any subclass of the super class.**](javascript:void(0);)

[**C - Variables, methods and constructors which are declared public in the superclass can be accessed by any class.**](javascript:void(0);)

**D - All of the above.**

**Answer : D**

**Explaination**

**All of the above are correct.**

**Q11 - What is runtime polymorphism?**

**A - Runtime polymorphism is a process in which a call to an overridden method is resolved at runtime rather than at compile-time.**

[**B - Runtime polymorphism is a process in which a call to an overloaded method is resolved at runtime rather than at compile-time.**](javascript:void(0);)

[**C - Both of the above.**](javascript:void(0);)

[**D - None of the above.**](javascript:void(0);)

**Answer : A**

**Explaination**

**Runtime polymorphism or dynamic method dispatch is a process in which a call to an overridden method is resolved at runtime rather than at compile-time. In this process, an overridden method is called through the reference variable of a superclass.**

**Q12 - Does garbage collection guarantee that a program will not run out of memory?**

[**A - True.**](javascript:void(0);)

**B - False.**

**Answer : B**

**Explaination**

**Garbage collection does not guarantee that a program will not run out of memory. It is possible for programs to use up memory resources faster than they are garbage collected. It is also possible for programs to create objects that are not subject to garbage collection**

**13. What is the output of the following program code?**

**abstract class C1{**

**public C1(){**

**System.out.print(1);**

**}**

**}**

**class C2 extends C1{**

**public C2(){**

**System.out.print(2);**

**}**

**}**

**class C3 extends C2{**

**public C3(){**

**System.out.println(3);**

**}**

**}**

**public class Test{**

**public static void main(String[] a){**

**new C3();**

**}**

**}**

**A.12**

**B.23**

**C.123**

**D.321**

**Solution: Option C**

**Q14. Which of these is correct about passing an argument by call-by-value process?  
a) Copy of argument is made into the formal parameter of the subroutine.  
b) Reference to original argument is passed to formal parameter of the subroutine.  
c) Copy of argument is made into the formal parameter of the subroutine and changes made on parameters of subroutine have effect on original argument.  
d) Reference to original argument is passed to formal parameter of the subroutine and changes made on parameters of subroutine have effect on original argument.**

**Answer: a  
Explanation: When we pass an argument by call-by-value a copy of argument is made into the formal parameter of the subroutine and changes made on parameters of subroutine have no effect on original argument, they remain the same.**

**Q15:What is the output of this program?**

**class string\_class {**

**public static void main(String args[])**

**{**

**String obj = "hello";**

**String obj1 = "world";**

**String obj2 = "hello";**

**System.out.println(obj.equals(obj1) + " " + obj.equals(obj2));**

**}**

**}**

**Falsefalse**

**truetrue**

**c)truefalse**

**falsetrue**

**Answer:d  
Explanation: equals() is method of class String, it is used to check equality of two String objects, if they are equal, true is retuned else false.  
output:  
$ javac string\_class.java  
$ java string\_class  
false true**

**Q16:Which of these classes implements Set interface?  
a) ArrayList  
b) HashSet  
c) LinkedList  
d) DynamicList**

**Answer: b  
Explanation: HashSet and TreeSet implements Set interface where as LinkedList and ArrayList implements List interface.**

**Q17:What is the output of this program?**

**class Output {**

**public static void main(String args[])**

**{**

**Object obj = new Object();**

**System.out.print(obj.getclass());**

**}**

**}**

**a)Object  
b)classObject  
c)classjava.lang.Object  
d)CompilationError**

**Answer:c  
Explanation:None.  
output:  
$javacOutput.java  
$javaOutput  
class java.lang.Object**

**Q18:Which of the following statements are incorrect?  
a) public members of class can be accessed by any code in the program.  
b) private members of class can only be accessed by other members of the class.  
c) private members of class can be inherited by a sub class, and become protected members in sub class.  
d) protected members of a class can be inherited by a sub class, and become private members of the sub class.**

**Answer:c  
Explanation: private members of a class can not be inherited by a sub class.**

**Q19:Which of these is a method of ObjectInput interface used to deserialize an object from a stream?  
a) int read()  
b) void close()  
c) Object readObject()  
d) Object WriteObject()  
View Answer**

**Answer: c  
Explanation: None.**

**Q20: Which of these method of Object class can clone an object?  
a)Objectcopy()  
b)copy()  
c)Objectclone()  
d)clone()**

**Answer:c**

**Q3:**

**What will be the output after running this java class??**

**Option 1:**

**Option 2:**

**Option 3:**

**Option 4: None of them.**

**Explanation:**

**Q3:**

**What will be the output after running this java class??**

**Option 1:**

**Option 2:**

**Option 3:**

**Option 4: None of them.**

**Explanation:**

**Q3:**

**What will be the output after running this java class??**

**Option 1:**

**Option 2:**

**Option 3:**

**Option 4: None of them.**

**Explanation:**

**Q3:**

**What will be the output after running this java class??**

**Option 1:**

**Option 2:**

**Option 3:**

**Option 4: None of them.**

**Explanation:**

**Q3:**

**What will be the output after running this java class??**

**Option 1:**

**Option 2:**

**Option 3:**

**Option 4: None of them.**

**Explanation:**

**Q3:**

**What will be the output after running this java class??**

**Option 1:**

**Option 2:**

**Option 3:**

**Option 4: None of them.**

**Explanation:**

**Q3:**

**What will be the output after running this java class??**

**Option 1:**

**Option 2:**

**Option 3:**

**Option 4: None of them.**

**Explanation:**

**Q3:**

**What will be the output after running this java class??**

**Option 1:**

**Option 2:**

**Option 3:**

**Option 4: None of them.**

**Explanation:**

**Q3:**

**What will be the output after running this java class??**

**Option 1:**

**Option 2:**

**Option 3:**

**Option 4: None of them.**

**Explanation:**

**Q3:**

**What will be the output after running this java class??**

**Option 1:**

**Option 2:**

**Option 3:**

**Option 4: None of them.**

**Explanation:**

**Q3:**

**What will be the output after running this java class??**

**Option 1:**

**Option 2:**

**Option 3:**

**Option 4: None of them.**

**Explanation:**

**Q3:**

**What will be the output after running this java class??**

**Option 1:**

**Option 2:**

**Option 3:**

**Option 4: None of them.**

**Explanation:**

**Q3:**

**What will be the output after running this java class??**

**Option 1:**

**Option 2:**

**Option 3:**

**Option 4: None of them.**

**Explanation:**

------------------------------------------------------------------------------------------------------------------------------------------