**Q1. What is the difference between an Inner Class and a Sub-Class?**

Ans: An Inner class is a class which is nested within another class. An Inner class has access rights for the class which is nesting it and it can access all variables and methods defined in the outer class.

A sub-class is a class which inherits from another class called super class. Sub-class can access all public and protected methods and fields of its super class.

**Q2. What are the various access specifiers for Java classes?**

Ans: In Java, access specifiers are the keywords used before a class name which defines the access scope. The types of access specifiers for classes are:

1. Public : Class,Method,Field is accessible from anywhere.

2. Protected:Method,Field can be accessed from the same class to which they belong or from the sub-classes,and from the class of same package,but not from outside.

3. Default: Method,Field,class can be accessed only from the same package and not from outside of it's native package.

4. Private: Method,Field can be accessed from the same class to which they belong.

**Q3. What's the purpose of Static methods and static variables?**

Ans: When there is a requirement to share a method or a variable between multiple objects of a class instead of creating separate copies for each object, we use static keyword to make a method or variable shared for all objects.

**Q4. What is data encapsulation and what's its significance?**

Ans: Encapsulation is a concept in Object Oriented Programming for combining properties and methods in a single unit.

Encapsulation helps programmers to follow a modular approach for software development as each object has its own set of methods and variables and serves its functions independent of other objects. Encapsulation also serves data hiding purpose.

**Q5. What is a singleton class? Give a practical example of its usage.**

A singleton class in java can have only one instance and hence all its methods and variables belong to just one instance. Singleton class concept is useful for the situations when there is a need to limit the number of objects for a class.

The best example of singleton usage scenario is when there is a limit of having only one connection to a database due to some driver limitations or because of any licensing issues.

**Q6. What are Loops in Java? What are three types of loops?**

Ans: Looping is used in programming to execute a statement or a block of statement repeatedly. There are three types of loops in Java:

1) For Loops

For loops are used in java to execute statements repeatedly for a given number of times. For loops are used when number of times to execute the statements is known to programmer.

2) While Loops

While loop is used when certain statements need to be executed repeatedly until a condition is fulfilled. In while loops, condition is checked first before execution of statements.

3) Do While Loops

Do While Loop is same as While loop with only difference that condition is checked after execution of block of statements. Hence in case of do while loop, statements are executed at least once.

**Q7: What is an infinite Loop? How infinite loop is declared?**

Ans: An infinite loop runs without any condition and runs infinitely. An infinite loop can be broken by defining any breaking logic in the body of the statement blocks.

Infinite loop is declared as follows:

```
for (;;)

{

// Statements to execute

// Add any loop breaking logic

}
```

**Q8. What is the difference between continue and break statement?**

Ans: break and continue are two important keywords used in Loops. When a break keyword is used in a loop, loop is broken instantly while when continue keyword is used, current iteration is broken and loop continues with next iteration.

In below example, Loop is broken when counter reaches 4.

```
for (counter=0;counter<10;counter++)
```

```
system.out.println(counter);

if (counter==4) {

break;}

}
```

In the below example when counter reaches 4, loop jumps to next iteration and any statements after the continue keyword are skipped for current iteration.

```
for (counter=0;counter<10;counter++)
system.out.println(counter);

if (counter==4) {

continue;

}

system.out.println("This will not get printed when counter is 4");

}
```

**Q9. What is the difference between double and float variables in Java?**

Ans: In java, float takes 4 bytes in memory while Double takes 8 bytes in memory. Float is single precision floating point decimal number while Double is double precision decimal number.

**Q10. What is Final Keyword in Java? Give an example.**

Ans: In java, a constant is declared using the keyword Final. Value can be assigned only once and after assignment, value of a constant can't be changed.

In below example, a constant with the name const_val is declared and assigned avalue:

```
Private Final int const_val=100
```

When a method is declared as final,it can NOT be overridden by the subclasses.This method are faster than any other method,because they are resolved at complied time.

When a class is declares as final,it cannot be subclassed. Example String,Integer and other wrapper classes.

**Q11. What is ternary operator? Give an example.**

Ans: Ternary operator , also called conditional operator is used to decide which value to assign to a variable based on a Boolean value evaluation. It's denoted as ?

In the below example, if rank is 1, status is assigned a value of "Done" else "Pending".

```
public class conditionTest {
 public static void main(String args[]) {
 String status;
 int rank = 3;
 status = (rank == 1) ? "Done" : "Pending";
 System.out.println(status);
 }
}
```

°C Celsius system 摄氏度
{ / } open/closed brace, open/closed curly 左/右花括号
( / ) open/close parenthesis, open/close paren 左/右圆括号;
() brakets/ parentheses 括号
[ / ] open/close bracket 左方括号; [] square brackets 方括号
. period, dot 句号，点
| vertical bar, vertical virgule 竖线
& ampersand, and, reference, ref 和，引用
* asterisk, multiply, star, pointer 星号，乘号，星，指针
/ slash, divide, oblique 斜线，斜杠，除号
// slash-slash, comment 双斜线，注释符 # pound 井号
\ backslash, sometimes escape 反斜线转义符, 有时表示转义符/续行符
~ tilde 波浪符
. full stop 句号
, comma 逗号
: colon 冒号
; semicolon 分号
? question mark 问号
! exclamation mark (英式英语) exclamation point (美式英语)
' apostrophe 撇号
– hyphen 连字号
-- dash 破折号
... dots / ellipsis 省略号
" single quotation marks 单引号
"" double quotation marks 双引号
|| parallel 双线号
& ampersand = and
～ swung dash 代字号
§ section; division 分节号
→ arrow 箭号;参见号

**Q12: How can you generate random numbers in Java?**

Ans:

- Using Math.random() you can generate random numbers in the range greater than or equal to 0.1 and less than 1.0
- Using Random class in package java.util

**Q13. What is default switch case? Give example.**

Ans: In a switch statement, default case is executed when no other switch condition matches. Default case is an optional case .
It can be declared only once all other switch cases have been coded.

In the below example, when score is not 1 or 2, default case is used.

```
public class switchExample {
int score=4;
public static void main(String args[]) {
switch (score) {
case 1:
system.out.println("Score is 1");
break;
case 2:
system.out.println("Score is 2");
break;
default:
system.out.println("Default Case");
}
```

```
}
}
```

**Q14. What's the base class in Java from which all classes are derived?**

Ans: java.lang.object

**Q15. Can main() method in Java can return any data?**

Ans: In java, main() method can't return any data and hence, it's always declared with a void return type.

**Q16. What are Java Packages? What's the significance of packages?**

Ans: In Java, package is a collection of classes and interfaces which are bundled together as they are related to each other. Use of packages helps developers to modularize the code and group the code for proper re-use. Once code has been packaged in Packages, it can be imported in other classes and used.

**Q17. Can we declare a class as Abstract without having any abstract method?**

Ans: Yes we can create an abstract class by using abstract keyword before class name even if it doesn't have any abstract method. However, if a class has even one abstract method, it must be declared as abstract otherwise it will give an error.

**Q18. What's the difference between an Abstract Class and Interface in Java?**

Ans: The primary difference between an abstract class and interface is that an interface can only possess declaration of public static methods with no concrete implementation while an abstract class can have members with any access specifiers (public, private etc) with or without concrete implementation.

Another key difference in the use of abstract classes and interfaces is that a class which implements an interface must implement all the methods of the interface while a class which inherits from an abstract class doesn't require implementation of all the methods of its super class.

A class can implement multiple interfaces but it can extend only one abstract class.

**Q19. What are the performance implications of Interfaces over abstract classes?**

Ans: Interfaces are slower in performance as compared to abstract classes as extra indirections are required for interfaces. Another key factor for developers to take into consideration is that any class can extend only one abstract class while a class can implement many interfaces.

Use of interfaces also puts an extra burden on the developers as any time an interface is implemented in a class; developer is forced to implement each and every method of interface.

**Q20. Does Importing a package imports its sub-packages as well in Java?**

Ans: In java, when a package is imported, its sub-packages aren't imported and developer needs to import them separately if required.

For example, if a developer imports a package university.*, all classes in the package named university are loaded but no classes from the sub-package are loaded. To load the classes from its sub-package ( say department), developer has to import it explicitly as follows:

Import university.department.*

**Q21. Can we declare the main method of our class as private?**

Ans: In java, main method must be public static in order to run any application correctly. If main method is declared as private, developer won't get any compilation error however, it will not get executed and will give a runtime error.

**Q22. How can we pass argument to a function by reference instead of pass by value?**

Ans: In java, we can pass argument to a function only by value and not by reference.

**Q23. How an object is serialized in java?**

Ans: In java, to convert an object into byte stream by serialization, an interface with the name Serializable is implemented by the class. All objects of a class implementing serializable interface get serialized and their state is saved in byte stream.

**Q24. When we should use serialization?**

Ans: Serialization is used when data needs to be transmitted over the network. Using serialization, object's state is saved and converted into byte stream .The byte stream is transferred over the network and the object is re-created at destination.

**Q25. Is it compulsory for a Try Block to be followed by a Catch Block in Java for Exception handling?**

Ans: Try block needs to be followed by either Catch block or Finally block or both. Any exception thrown from try block needs to be either caught in the catch block or else any specific tasks to be performed before code abortion are put in the Finally block.

**Q26. Is there any way to skip Finally block of exception even if some exception occurs in the exception block?**

Ans: If an exception is raised in Try block, control passes to catch block if it exists otherwise to finally block. Finally block is always executed when an exception occurs and the only way to avoid execution of any statements in Finally block is by aborting the code forcibly by writing following line of code at the end of try block:

System.exit(0);

**Q27. When the constructor of a class is invoked?**

Ans: The constructor of a class is invoked every time an object is created with new keyword.

For example, in the following class two objects are created using new keyword and hence, constructor is invoked two times.

```
public class const_example {

const_example() {

system.out.println("Inside constructor");

}

public static void main(String args[]) {

const_example c1=new const_example();

const_example c2=new const_example();

}

}
```

**Q28. Can a class have multiple constructors?**

Ans: Yes, a class can have multiple constructors with different parameters. Which constructor gets used for object creation depends on the arguments passed while creating the objects.

**Q29. Can we override static methods of a class?**

Ans: We cannot override static methods. Static methods belong to a class and not to individual objects and are resolved at the time of compilation (not at runtime).Even if we try to override static method,we will not get an complitaion error,nor the impact of overriding when running the code.

**Q30. In the below example, what will be the output?**

```
public class superclass {
```

```
public void displayResult() {

system.out.println("Printing from superclass");

}

}

public class subclass extends superclass {

public void displayResult() {

system.out.println("Displaying from subClass");

super.displayResult();

}

public static void main(String args[]) {

subclass obj=new subclass();

obj.displayResult();

}

}
```

**Ans**: Output will be:

Displaying from subclass

Displaying from superclass

**Q31. Is String a data type in java?**

Ans: String is not a primitive data type in java. When a string is created in java, it's actually an object of Java.Lang.String class that gets created. After creation of this string object, all built-in methods of String class can be used on the string object.

**Q32. In the below example, how many String Objects are created?**

```
String s1="I am Java Expert";

String s2="I am C Expert";
```

String s3="I am Java Expert";

Ans: In the above example, two objects of Java.Lang.String class are created. s1 and s3 are references to same object.

**Q33. Why Strings in Java are called as Immutable?**

Ans: In java, string objects are called immutable as once value has been assigned to a string, it can't be changed and if changed, a new object is created.

In below example, reference str refers to a string object having value "Value one".

String str="Value One";

old String obj still exists, but no reference.

When a new value is assigned to it, a new String object gets created and the reference is moved to the new object.

str="New Value";

**Q34. What's the difference between an array and Vector?**

Ans: An array groups data of same primitive type and is static in nature while vectors are dynamic in nature and can hold data of different data types.

**Q35. What is multi-threading?**

Ans: Multi threading is a programming concept to run multiple tasks in a concurrent manner within a single program. Threads share same process stack and running in parallel. It helps in performance improvement of any program.

**Q36. Why Runnable Interface is used in Java?**

Ans: Runnable interface is used in java for implementing multi threaded applications. Java.Lang.Runnable interface is implemented by a class to support multi threading.

**Q37. What are the two ways of implementing multi-threading in Java?**

Ans: Multi threaded applications can be developed in Java by using any of the following two methodologies:

1. By using Java.Lang.Runnable Interface. Classes implement this interface to enable multi threading. There is a Run() method in this interface which is implemented.

2. By writing a class that extend Java.Lang.Thread class.

**Q38. When a lot of changes are required in data, which one should be a preference to be used? String or StringBuffer?**

Ans: Since StringBuffers are dynamic in nature and we can change the values of StringBuffer objects unlike String which is immutable, it's always a good choice to use StringBuffer when data is being changed too much. If we use String in such a case, for every data change a new String object will be created which will be an extra overhead.

**Q39. What's the purpose of using Break in each case of Switch Statement?**

Ans: Break is used after each case (except the last one) in a switch so that code breaks after the valid case and doesn't flow in the proceeding cases too.

If break isn't used after each case, all cases after the valid case also get executed resulting in wrong results.

**Q40. How garbage collection is done in Java?**

Ans: In java, when an object is not referenced any more, garbage collection takes place and the object is destroyed automatically. For automatic garbage collection java calls either System.gc() method or Runtime.gc() method.

**Q41. How we can execute any code even before main method?**

Ans: If we want to execute any statements before even creation of objects at load time of class, we can use a static block of code in the class. Any statements inside this static block of code will get executed once at the time of loading the class even before creation of objects in the main method.

**Q42. Can a class be a super class and a sub-class at the same time? Give example.**

Ans: If there is a hierarchy of inheritance used, a class can be a super class for another class and a sub-class for another one at the same time.

In the example below, continent class is sub-class of world class and it's super class of country class.

```
public class world {

.....

}

public class continenet extends world {

.....
```

}

public class country extends continent {

.....

}

**Q43. How objects of a class are created if no constructor is defined in the class?**

Ans: Even if no explicit constructor is defined in a java class, objects get created successfully as a default constructor is implicitly used for object creation. This constructor has no parameters.

**Q44. In multi-threading how can we ensure that a resource isn't used by multiple threads simultaneously?**

Ans: In multi-threading, access to the resources which are shared among multiple threads can be controlled by using the concept of synchronization. Using synchronized keyword, we can ensure that only one thread can use shared resource at a time and others can get control of the resource only once it has become free from the other one using it.

**Q45. Can we call the constructor of a class more than once for an object?**

Ans: Constructor is called automatically when we create an object using new keyword. It's called only once for an object at the time of object creation and hence, we can't invoke the constructor again for an object after its creation.

**Q46. There are two classes named classA and classB. Both classes are in the same package. Can a private member of classA can be accessed by an object of classB?**

Ans: Private members of a class aren't accessible outside the scope of that class and any other class even in the same package can't access them.

**Q47. Can we have two methods in a class with the same name?**

Ans: We can define two methods in a class with the same name but with different number/type of parameters. Which method is to get invoked will depend upon the parameters passed.

For example in the class below we have two print methods with same name but different parameters. Depending upon the parameters, appropriate one will be called:

public class methodExample {

public void print() {

```
system.out.println("Print method without parameters.");

}

public void print(String name) {

system.out.println("Print method with parameter");

}

public static void main(String args[]) {

methodExample obj1= new methodExample();

obj1.print();

obj1.print("xx");

}

}
```

**Q48. How can we make copy of a java object?**

Ans: We can use the concept of cloning to create copy of an object. Using clone, we create copies with the actual state of an object.

Clone() is a method of Cloneable interface and hence, Cloneable interface needs to be implemented for making object copies.

**Q49. What's the benefit of using inheritance?**

Ans: Key benefit of using inheritance is reusability of code as inheritance enables sub-classes to reuse the code of its super class. Polymorphism (Extensibility ) is another great benefit which allow new functionality to be introduced without effecting existing derived classes.

**Q50. What's the default access specifier for variables and methods of a class?**

Ans: Default access specifier for variables and method is package protected i.e variables and class is available to any other class but in the same package,not outside the package.

**Q51. Give an example of use of Pointers in Java class.**

Ans: There are no pointers in Java. So we can't use concept of pointers in Java.

**Q52. How can we restrict inheritance for a class so that no class can be inherited from it?**

Ans: If we want a class not to be extended further by any class, we can use the keyword **Final** with the class name.

In the following example, Stone class is Final and can't be extend

```
public Final Class Stone {

// Class methods and Variables

}
```

**Q53. What's the access scope of Protected Access specifier?**

Ans: When a method or a variable is declared with Protected access specifier, it becomes accessible in the same class,any other class of the same package as well as a sub-class.

| Access Levels | | | | |
|---|---|---|---|---|
| **Modifier** | **Class** | **Package** | **Subclass** | **World** |
| public | Y | Y | Y | Y |
| protected | Y | Y | Y | N |
| no modifier | Y | Y | N | N |
| private | Y | N | N | N |

**Q54. What's difference between Stack and Queue?**

Ans: Stack and Queue both are used as placeholder for a collection of data. The primary difference between a stack and a queue is that stack is based on Last in First out (LIFO) principle while a queue is based on FIFO (First In First Out) principle.

**Q55. In java, how we can disallow serialization of variables?**

Ans: If we want certain variables of a class not to be serialized, we can use the keyword **transient** while declaring them. For example, the variable trans_var below is a transient variable and can't be serialized:

```
public class transientExample {

private transient trans_var;

// rest of the code
```

}

**Q56. How can we use primitive data types as objects?**

Ans: Primitive data types like int can be handled as objects by the use of their respective wrapper classes. For example, Integer is a wrapper class for primitive data type int. We can apply different methods to a wrapper class, just like any other object.

**Q57. Which types of exceptions are caught at compile time?**

Ans: Checked exceptions can be caught at the time of program compilation. Checked exceptions must be handled by using try catch block in the code in order to successfully compile the code.

**Q58. Describe different states of a thread.**

Ans: A thread in Java can be in either of the following states:

- Ready: When a thread is created, it's in Ready state.
- Running: A thread currently being executed is in running state.
- Waiting: A thread waiting for another thread to free certain resources is in waiting state.
- Dead: A thread which has gone dead after execution is in dead state.

**Q59. Can we use a default constructor of a class even if an explicit constructor is defined?**

Ans: Java provides a default no argument constructor if no explicit constructor is defined in a Java class. But if an explicit constructor has been defined, default constructor can't be invoked and developer can use only those constructors which are defined in the class.

**Q60. Can we override a method by using same method name and arguments but different return types?**

Ans: The basic condition of method overriding is that method name, arguments as well as return type must be exactly same as is that of the method being overridden. Hence using a different return type doesn't override a method.

**Q61.What will be the output of following piece of code?**

```
public class operatorExample {

public static void main(String args[]) {

int x=4;

system.out.println(x++);

}
```

}

Ans: In this case postfix ++ operator is used which first returns the value and then increments. Hence it's output will be 4.

**Q61. A person says that he compiled a java class successfully without even having a main method in it? Is it possible?**

Ans: main method is an entry point of Java class and is required for execution of the program however; a class gets compiled successfully even if it doesn't have a main method. It can't be run though.

belong to object belong to class

**Q62. Can we call a non-static method from inside a static method?**

Ans: Non-Static methods are owned by objects of a class and have object level scope and in order to call the non-Static methods from a static block (like from a static main method), an object of the class needs to be created first. Then using object reference, these methods can be invoked.

**Q63. What are the two environment variables that must be set in order to run any Java programs?**

Ans: Java programs can be executed in a machine only once following two environment variables have been properly set:

1. PATH variable
2. CLASSPATH variable

**Q64. Can variables be used in Java without initialization?**

Ans: In Java, if a variable is used in a code without prior initialization by a valid value, program doesn't compile and gives an error as no default value is assigned to variables in Java.

**Q65. Can a class in Java be inherited from more than one class?**

Ans: In Java, a class can be derived from only one class and not from multiple classes. Multiple inheritances is not supported by Java.

**Q66. Can a constructor have different name than a Class name in Java?**

Ans: Constructor in Java must have same name as the class name and if the name is different, it doesn't act as a constructor and compiler thinks of it as a normal method.

**Q67. What will be the output of Round(3.7) and Ceil(3.7)?**

Ans: Round(3.7) returns 4 and Ceil(3.7) returns 4.

**Q68: Can we use goto in Java to go to a particular line?**

Ans: In Java, there is not goto keyword and java doesn't support this feature of going to a particular labeled line.

**Q69. Can a dead thread be started again?**

Ans: In java, a thread which is in dead state can't be started again. There is no way to restart a dead thread.

**Q70. Is the following class declaration correct?**

**Ans:**

```
public abstract final class testClass {

// Class methods and variables

}
```

Ans: The above class declaration is incorrect as an abstract class can't be declared as Final.

**Q71. Is JDK required on each machine to run a Java program?**

JDK：运行环境+开发环境；JRE：运行环境

Ans: JDK is development Kit of Java and is required for development only and to run a Java program on a machine, JDK isn't required. Only JRE is required.

**Q72. What's the difference between comparison done by equals method and == operator?**

Ans: In Java, equals() method is used to compare the contents of two string objects and returns true if the two have same value while == operator compares the references of two string objects.

In the following example, equals() returns true as the two string objects have same values. However == operator returns false as both string objects are referencing to different objects:

```
public class equalsTest {

   public static void main(String args[]) {

      String str1 = new String("Hello World");

      String str2 = new String("Hello World");

      if (str1.equals(str2))

      { // this condition is true
```

```
        System.out.println("str1 and str2 are equal in terms of values");
    }
    if (str1 == str2) {
        //This condition is true
        System.out.println("Both strings are referencing same object");
    } else
    {
        // This condition is NOT true
        System.out.println("Both strings are referencing different objects");
    }
  }
}
```

**Q73. Is it possible to define a method in Java class but provide it's implementation in the code of another language like C?**

Ans: Yes, we can do this by use of native methods. In case of native method based development, we define public static methods in our Java class without its implementation and then implementation is done in another language like C separately.

**Q74. How are destructors defined in Java?**

Ans: In Java, there are no destructors defined in the class as there is no need to do so. Java has its own garbage collection mechanism which does the job automatically by destroying the objects when no longer referenced.

**Q75. Can a variable be local and static at the same time?**

Ans: No a variable can't be static as well as local at the same time. Defining a local variable as static gives compilation error.

**Q76. Can we have static methods in an Interface?**

Ans: Static methods can't be overridden in any class while any methods in an interface are by default abstract and are supposed to be implemented in the classes being implementing the interface. So it makes no sense to have static methods in an interface in Java.

**Q77. In a class implementing an interface, can we change the value of any variable defined in the interface?**

Ans: No, we can't change the value of any variable of an interface in the implementing class as all variables defined in the interface are by default public, static and Final and final variables are like constants which can't be changed later.

**Q78. Is it correct to say that due to garbage collection feature in Java, a java program never goes out of memory?**

Ans: Even though automatic garbage collection is provided by Java, it doesn't ensure that a Java program will not go out of memory as there is a possibility that creation of Java objects is being done at a faster pace compared to garbage collection resulting in filling of all the available memory resources.

So, garbage collection helps in reducing the chances of a program going out of memory but it doesn't ensure that.

**Q79. Can we have any other return type than void for main method?**

Ans: No, Java class main method can have only void return type for the program to get successfully executed.

Nonetheless , if you absolutely must return a value to at the completion of main method , you can use System.exit(int status)

**Q80. I want to re-reach and use an object once it has been garbage collected. How it's possible?**

Ans: Once an object has been destroyed by garbage collector, it no longer exists on the heap and it can't be accessed again. There is no way to reference it again.

**Q81. In Java thread programming, which method is a must implementation for all threads?**

Ans: Run() is a method of Runnable interface that must be implemented by all threads.

**Q82. I want to control database connections in my program and want that only one thread should be able to make database connection at a time. How can I implement this logic?**

Ans: This can be implemented by use of the concept of synchronization. Database related code can be placed in a method which hs **synchronized** keyword so that only one thread can access it at a time.

**Q83. How can an exception be thrown manually by a programmer?**

Ans: In order to throw an exception in a block of code manually, **throw** keyword is used. Then this exception is caught and handled in the catch block.

```
public void topMethod(){
try{
excMethod();
}catch(ManualException e){ }
}

public void excMethod{
String name=null;
if(name == null){
throw (new ManualException("Exception thrown manually ");
}
}
```

**Q84. I want my class to be developed in such a way that no other class (even derived class) can create its objects. How can I do so?**

Ans: If we declare the constructor of a class as private, it will not be accessible by any other class and hence, no other class will be able to instantiate it and formation of its object will be limited to itself only.

**Q85. How objects are stored in Java?**

Ans: In java, each object when created gets a memory space from a heap. When an object is destroyed by a garbage collector, the space allocated to it from the heap is re-allocated to the heap and becomes available for any new objects.

**Q86. How can we find the actual size of an object on the heap?**

Ans: In java, there is no way to find out the exact size of an object on the heap.

**Q87. Which of the following classes will have more memory allocated?**

**Class A: Three methods, four variables, no object**

**Class B: Five methods, three variables, no object**

Ans: Memory isn't allocated before creation of objects. Since for both classes, there are no objects created so no memory is allocated on heap for any class.

**Q88. What happens if an exception is not handled in a program?**

Ans: If an exception is not handled in a program using try catch blocks, program gets aborted and no statement executes after the statement which caused exception throwing.

**Q89. I have multiple constructors defined in a class. Is it possible to call a constructor from another constructor's body?**

Ans: If a class has multiple constructors, it's possible to call one constructor from the body of another one using **this()**.

**Q90. What's meant by anonymous class?**

Ans: An anonymous class is a class defined without any name in a single line of code using new keyword.

For example, in below code we have defined an anonymous class in one line of code:

```
public java.util.Enumeration testMethod()

{

return new java.util.Enumeration()

{

@Override

public boolean hasMoreElements()

{

// TODO Auto-generated method stub

return false;

}

@Override

public Object nextElement()

{

// TODO Auto-generated method stub

return null;

}

}
```

**Q91. Is there a way to increase the size of an array after its declaration?**

Ans: Arrays are static and once we have specified its size, we can't change it. If we want to use such collections where we may require a change of size ( no of items), we should prefer vector over array.

**Q92. If an application has multiple classes in it, is it okay to have a main method in more than one class?**

Ans: If there is main method in more than one classes in a java application, it won't cause any issue as entry point for any application will be a specific class and code will start from the main method of that particular class only.

**Q93. I want to persist data of objects for later use. What's the best approach to do so?**

Ans: The best way to persist data for future use is to use the concept of serialization.

**Q94. What is a Local class in Java?**

Ans: In Java, if we define a new class inside a particular block, it's called a local class. Such a class has local scope and isn't usable outside the block where its defined.

**Q95. String and StringBuffer both represent String objects. Can we compare String and StringBuffer in Java?**

Ans: Although String and StringBuffer both represent String objects, we can't compare them with each other and if we try to compare them, we get an error.

**Q96. Which API is provided by Java for operations on set of objects?**

Ans: Java provides a Collection API which provides many useful methods which can be applied on a set of objects. Some of the important classes provided by Collection API include ArrayList, HashMap, TreeSet and TreeMap.

**Q97. Can we cast any other type to Boolean Type with type casting?**

Ans: No, we can neither cast any other primitive type to Boolean data type nor can cast Boolean data type to any other primitive data type.

**Q98. Can we use different return types for methods when overridden?**

Ans: The basic requirement of method overriding in Java is that the overridden method should have same name, and parameters.But a method can be overridden with a different return type as long as the new return type extends the original.

For example , method is returning a reference type.

Class B extends A{

A method(int x){

//original method

}

B method(int x){

//overridden method

}

}

**Q99. What's the base class of all exception classes?**

Ans: In Java, **Java.lang.Throwable** is the super class of all exception classes and all exception classes are derived from this base class.

**Q100. What's the order of call of constructors in inheritiance?**

Ans: In case of inheritance, when a new object of a derived class is created, first the constructor of the super class is invoked and then the constructor of the derived class is invoked.

Guru99 Provides FREE ONLINE TUTORIAL on Various courses like

| | | | | |
|---|---|---|---|---|
| Java | MIS | MongoDB | BigData | Cassandra |
| Web Services | SQLite | JSP | Informatica | Accounting |
| SAP Training | Python | Excel | ASP Net | HBase |
| Project Management | Test Management | Business Analyst | Ethical Hacking | PMP |
| Live Project | SoapUI | Photoshop | Manual Testing | Mobile Testing |
| Data Warehouse | R Tutorial | Tableau | DevOps | AWS |
| Jenkins | Agile Testing | RPA | JUnit | Software Engineering |
| Selenium | CCNA | AngularJS | NodeJS | PLSQL |