ASSGNMENT NO\_3

**1. What is Java?**

=>Java is programming Language.A distributed application is one which always runs in the context of Browser or World Wide

Web. All distributed applications can be accessed across the globe. For example JAVA and DOT NET.

JAVA always provides a facility called server independent, platform independent language.

JAVA supports a concept called design patterns (design patterns are predefined proved rules by industry experts to avoid side effects [recurring problems] which are occurring in software development).

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

**2. What is a package in Java? List down various advantages of packages**.

=>Package in Java is a mechanism to encapsulate a group of classes, sub packages and

interfaces. Packages are used for:

Preventing naming conflicts. For example there can be two classes with name Employee in

two packages, college.staff.cse.Employee and college.staff.ee.Employee Making searching/locating

and usage of classes, interfaces, enumerations and annotations easier Providing controlled access: protected and default have package level access control.

A protected member is accessible by classes in the same package and its subclasses.

A default member (without any access specifier) is accessible by classes in the same package only.

Packages can be considered as data encapsulation (or data-hiding).

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

**3. Explain JDK, JRE and JVM?**

=>JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a

software development environment which is used to develop Java applications and applets.

It physically exists. It contains JRE + development tools.

=>JRE is an acronym for Java Runtime Environment. It is also written as Java RTE. The Java Runtime Environment is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It physically exists. It contains a set of libraries + other files that JVM uses at runtime.

=>JVM (Java Virtual Machine) is an abstract machine. It is called a virtual machine because it doesn't physically exist. It is a specification that provides a runtime environment in which Java bytecode can be executed. It can also run those programs which are written in other languages and compiled to Java bytecode.

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

**4. Explain public static void main(String args[]) in Java.**

=>1. public- Here public is an access specifier which allows thhe main method to be accessble everywhere.

2. static- static helps main method to get loaded without getting alled by any instance/object.

3. void- void clarifies that the main method will not return any value.

4. main- It's the name of the method.

5. String[] args- Here we are defining a String array to pass arguments at command line.

args is the variable name of the String array. It can be changed to anything such as String [] a.

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

**5. What are the differences between C++ and Java?**

=>C++ and Java both are object-oriented programming languages.

Yet, both languages differ from each other in many ways.

C++ is derived from C and has the features of both procedural and object-oriented

programming languages. C++ was designed for application and System development.

Java is built upon a virtual machine which is very secure and highly portable in nature.

It is grouped with a comprehensive library to provide support for the abstraction of the

existing platform.

Java was mainly designed for application programming and has a functionality

of an interpreter for printing systems which was later developed into network computing.

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

**6. Why Java is platform independent?**

=>Using the Java Virtual Machine we can make the byte code understandable to any platform.

That is why the byte code is known as platform-independent. But on the other hand

the Java Virtual Machine is different for each platform; that is why it is known as

platform-dependent. Java is platform-independent because it does not depend on any type

of platform. Hence, Java is platform-independent language.

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

**7. What are wrapper classes in Java?**

=>A Wrapper class is a class whose object wraps or contains primitive data types.When we create an object to a wrapper class, it contains a field and in this field,we can store primitive data types. In other words, we can wrap a primitive value into a wrapper class object.

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

**8. Why pointers are not used in Java?**

=>Java doesn’t support pointer explicitly, But java uses pointer implicitly,Java use pointers

for manipulations of references but these pointers are not available for outside use.

Any operations implicitly done by the language are actually NOT visible.

Java doesn’t have pointers (in the C/C++ sense) because it doesn’t need them for general

purpose OOP programming. Furthermore, adding pointers to Java would undermine

security and robustness and make the language more complex.

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

**9. List some features of Java?**

=>Platform Independent

Simple

Secure

Architecture-neutral

Portable

Robust

Multithreaded

Dynamic

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

**10. Why is Java Architectural Neutral?**

=>It’s compiler generates an architecture-neutral object file format, which makes the

compiled code to be executable on many processors, with the presence of Java runtime system.

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

**11. How Java enabled High Performance?**

=>Java uses Just-In-Time compiler to enable high performance. Just-In-Time compiler is a

program that turns Java bytecode, which is a program that contains instructions that must

be interpreted into instructions that can be sent directly to the processor.

**12. Why Java is considered dynamic?**

=>Java - Inheritance

Java - Overriding

Java - Polymorphism

Java - Abstraction

Java - Encapsulation

Java - Interfaces

Java - Packages

Java Advanced

Java - Data Structures

Java - Collections

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

**13.What is Java Virtual Machine and how it is considered in context of Java’s platform independent feature?**

Java Virtual Machine (JVM) is a specification that provides runtime environment in which java bytecode(. class files) can be executed. The JVM is the platform. ... JVM makes this possible because it is aware of the specific instruction lengths and other particularities of the platform (Operating System).

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

**14.List two Java IDE’s?**

The following are the best Java IDEs that are mostly used in the world:

* [Eclipse](https://www.javatpoint.com/java-ides#Eclipse)
* NetBeans

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

**15.Why Java is called as “Platform” ?**

Platform is a software and hardware programs that runs. JAVA is platform independent because it having its own JVM so that it can run on any platform . java is platform independent , which means once written you can run it anywhere. The platform is a hardware or software used to run an application.

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

**16.Is Java Pure-Object oriented Language ?**

Java is not an pure object oriented language because it supports Primitive datatype such as int, byte, long... etc, to be used, which are not objects. Contrast with a pure OOP language like Smalltalk, where there are no primitive types, and boolean, int and methods are all objects. ... All user defined types are objects.

Java language is not a Pure Object Oriented Language as it contain these properties: Primitive Data Type ex. ... In Smalltalk, primitive values such as integers, booleans and characters are also objects. In Java, we have predefined types as non-objects (primitive types).

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

**17.Which version of java have u learned? Name some of the new features added to it.**

Oracle released a new version of Java as Java 8 in March 18, 2014. It was a revolutionary release of the Java for software development platform. It includes various upgrades to the Java programming, JVM, Tools and libraries.

Java 8 provides following features for Java Programming:

* Lambda expressions,
* Method references,
* Functional interfaces,
* Stream API,
* Default methods,
* Base64 Encode Decode,
* Static methods in interface,
* Optional class,
* Collectors class,
* ForEach() method,
* Parallel array sorting,
* Nashorn JavaScript Engine,
* Parallel Array Sorting,
* Type and Repating Annotations,
* IO Enhancements,

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

**18.What gives Java its 'write once and run anywhere' nature?**

The "Write Once, Run Everywhere" slogan refers to the fact that an application written is Java can be run on any hardware which has the Java Virtual Machine (JVM), and that the JVM is now licensed to hundreds of operating systems vendors systems including Microsoft for Windows.

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

**19.Difference between path and classpath.**

Let us clear the difference in points:

PATH

a) An environment variable which is used by the operating system to find the executables.

b) PATH is nothing but setting up an environment for operating system. Operating System will look in this PATH for executables.

c) Refers to the system

CLASSPATH

a) An environment variable which is used by the Java compiler to find the path, of classes i.e in J2EE we give the path of jar files.

b) Classpath is nothing but setting up the environment for Java. Java will use to find compiled classes.

c) Refers to the Developing Enviornment.

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

**20.What is the signature of main function in java ?**

The method signature for the main() method contains three modifiers: public indicates that the main() method can be called by any object. static indicates that the main() method is a class method. void indicates that the main() method has no return value.

Void keyword acknowledges the compiler that main() method does not return any value. main(): It is a default signature which is predefined in the JVM. It is called by JVM to execute a program line by line and end the execution after completion of this method. We can also overload the main() method.

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

**21.What is the difference between JDK and JRE?**

JDK is a software development kit whereas JRE is a software bundle that allows Java program to run, whereas JVM is an environment for executing bytecode. The full form of JDK is Java Development Kit, while the full form of JRE is Java Runtime Environment, while the full form of JVM is Java Virtual Machine.

JDK includes the JRE plus command-line development tools such as compilers and debuggers that are necessary or useful for developing applets and applications. JRE is basically the Java Virtual Machine where your Java programs run on. JDK is an abstract machine.

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

**22.What is JVM ? What it does?**

A Java virtual machine (JVM) is a virtual machine that enables a computer to run Java programs as well as programs written in other languages that are also compiled to Java bytecode. ... The JVM reference implementation is developed by the OpenJDK project as open source code and includes a JIT compiler called HotSpot.

The JVM has two primary functions: to allow Java programs to run on any device or operating system (known as the "Write once, run anywhere" principle), and to manage and optimize program memory.

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

**23.Why JVM is called as “virtual machine”?**

The JVM is "virtual" because it is generally implemented in software on top of a "real" hardware platform and operating system. All Java programs are compiled for the JVM. Therefore, the JVM must be implemented on a particular platform before compiled Java programs will run on that platform.

**24.What are the main components of JVM? Explain them. Or Explain JVM Architecture.**

It performs three major functions viz. Loading, Linking, and Initialization. JVM Method Area stores class structures like metadata, the constant runtime pool, and the code for methods. All the Objects, their related instance variables, and arrays are stored in the heap.

Java Virtual Machine (JVM) is a engine that provides runtime environment to drive the Java Code or applications. It converts Java bytecode into machines language. JVM is a part of Java Run Environment (JRE). In other programming languages, the compiler produces machine code for a particular system.

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

**25.What is the difference between Java compiler ( javac ) and JIT ?**

When compiling a java program, the static compiler that is run using the command javac converts the source code to byte code which are in the form of . class files. ... JIT compiles the code when it is needed but not before runtime.

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

**26.Is Empty .java file name a valid source file name?**

Since, you cannot leave class name empty as well as you can't also change its name to any other since it is public. ... If you write a file in Java which is already present in the location, it will be overwritten automatically. Unless you are writing to that file with an append flag set to True.

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

**27. Is JRE different for different Platforms ?**

Whenever we try to run the code, JVM requires some library set and files for code execution and these files are presented in JRE. JRE = JVM + set of libraries. ... JRE is also platform dependent. That means we have different JRE versions for different platforms.

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

**28.Difference between C++ and java in terms of object creation.**

C++ supports manual object management with the help of new and delete keywords whereas Java has built-in automatic garbage collection. C++ supports structures whereas Java doesn't supports structures. C++ supports unions while Java doesn't support unions.

**29.Who invokes main() function ?**

In 'C', the "main" function is called by the operating system when the user runs the program and it is treated the same way as every function, it has a return type. Although you can call the main() function within itself and it is called recursion.

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

**30.What is .class file known as ?**

 A Java class file is a file (with the . class filename extension) containing Java bytecode that can be executed on the Java Virtual Machine (JVM). A Java class file is usually produced by a Java compiler from Java programming language source files.

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

**31.Can we define more than one public class in a java source code ? what is the rule of public class and file name?**

No, while defining multiple classes in a single Java file you need to make sure that only one class among them is public. If you have more than one public classes a single file a compile-time error will be generated.

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

**32.What is JIT compiler?**

The Just-In-Time (JIT) compiler is a component of the runtime environment that improves the performance of Java™ applications by compiling bytecodes to native machine code at run time. ... The JIT compiler helps improve the performance of Java programs by compiling bytecodes into native machine code at run time.

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

**33.How many types of memory areas are allocated by JVM?**

The memory in the JVM divided into 5 different parts:

* Class(Method) Area.
* Heap.
* Stack.
* Program Counter Register.
* Native Method Stack.

**34.What is the rule for local member in java.**

Local variables cannot use any of the access level since their scope is only inside the method. Final is the Only Non Access Modifier that can be applied to a local variable. Local variables are not assigned a default value, hence they need to be initialized.

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

**35.What are the various access specifiers in Java?**

Access Specifiers in Java | Access Modifiers

* Public Access Specifier.
* Protected Access Specifier.
* Default Access Specifier.
* Private Access Specifiers.

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

**36.What is the rule for local member in java.**

Local variables cannot use any of the access level since their scope is only inside the method. Final is the Only Non Access Modifier that can be applied to a local variable. Local variables are not assigned a default value, hence they need to be initialized.

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

**37.What is native code?**

Native code is computer programming (code) that is compiled to run with a particular processor (such as an Intel x86-class processor) and its set of instructions. ... Java bytecode and Microsoft's Intermediate Language can be compiled into native code before execution by a just-in-time compiler for faster performance.

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

**38.Why there is no sizeof operator in java ?**

Because the size of primitive types is explicitly mandated by the Java language. There is no variance between JVM implementations. Moreover, since allocation is done by the new operator depending on its argument there is no need to specify the amount of memory needed.

**39.What kinds of programs u can develop using Java ?**

* Mobile Applications
* Desktop GUI Applications
* Web-based Applications
* Enterprise Applications
* Scientific Applications
* Gaming Applications
* Big Data technologies
* Business Applications
* Distributed Applications
* Cloud-based Applications

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

**40.U have reference type as a member of class. What is the default value it gets?**

‘null’The default value of a reference type variable is null when they are not initialized. Null means not refering to any object.

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

**41.What is the job done by classloader ?**

The Java ClassLoader is a part of the Java Runtime Environment that dynamically loads Java classes into the Java Virtual Machine. The Java run time system does not need to know about files and file systems because of classloaders. Java classes aren't loaded into memory all at once, but when required by an application.

**42.Explain the hierarchy of classloaders in java.**

ClassLoader is hierarchical in loading a class into memory. Whenever a request is raised to load a class, it delegates it to the parent classloader. This is how uniqueness is maintained in the runtime environment. If the parent class loader doesn’t find the class then the class loader itself tries to load the class.

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

**43.What is the role played by Bytecode Verifier ?**

The bytecode verifier acts as a sort of gatekeeper: it ensures that code passed to the Java interpreter is in a fit state to be executed and can run without fear of breaking the Java interpreter. Imported code is not allowed to execute by any means until after it has passed the verifier's tests.

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

**44.What are the memory areas allocated by JVM ?**

The memory in the JVM divided into 5 different parts:

* Class(Method) Area.
* Heap.
* Stack.
* Program Counter Register.
* Native Method Stack.

Heap − Runtime storage allocation for objects (reference types). Stack − Storage for local variables and partial results. A stack contains frames and allocates one for each thread. ... Native method stacks − It contains all the native methods used by the application.

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

**45.What kinds of programs u can develop using Java?**

* Mobile Applications
* Desktop GUI Applications
* Web-based Applications
* Enterprise Applications
* Scientific Applications
* Gaming Applications
* Big Data technologies
* Business Applications
* Distributed Applications
* Cloud-based Applications

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

**46.When parseInt() method can be used?**

The Integer. parseInt() java method is used primarily in parsing a String method argument into an Integer object. The Integer object is a wrapper class for the int primitive data type of java API.

 Convert a string to an integer with the parseInt method of the Java Integer class. The parseInt method is to convert the String to an int and throws a NumberFormatException if the string cannot be converted to an int type.

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

**47.What is finalized() method ?**

Finalize() is the method of Object class. This method is called just before an object is garbage collected. finalize() method overrides to dispose system resources, perform clean-up activities and minimize memory leaks.

The finalize() method of Object class is a method that the Garbage Collector always calls just before the deletion/destroying the object which is eligible for Garbage Collection, so as to perform clean-up activity.

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

**48.Difference between C++ pointer and Java reference.**

Reference: A reference is a variable that refers to something else and can be used as an alias for that something else. ... Pointers are a particular implementation of the concept of a reference, and the term tends to be used only for languages that give you direct access to the memory address. References are used to refer an existing variable in another name whereas pointers are used to store address of variable. References cannot have a null value assigned but pointer can. A reference variable can be referenced by pass by value whereas a pointer can be referenced by pass by reference.

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

**49.U have reference type as a member of class. What is the default value it gets?**

‘null’ The default value of a reference type variable is null when they are not initialized. Null means not refering to any object.

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

**50.What are the expressions allowed in switch block of java ?**

A switch works with the byte , short , char , and int primitive data types. It also works with enumerated types (discussed in Enum Types), the String class, and a few special classes that wrap certain primitive types: Character , Byte , Short , and Integer (discussed in Numbers and Strings).