

EDAC ASSIGNMENT 3

1. What is Java?

Java is a general purpose, class-based object-oriented programming designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast secure, and reliable; therefore, it is widely used for developing Java applications in laptops, data centers, gaming consoles, scientific supercomputers, cellphones, etc.

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

A **java package** is a group of similar types of classes, interfaces and sub-packages.

Package in java can be categorized in two form, built-in package and user-defined package.

There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Advantages of Java packages

- 1) Java package is used to categorize the classes and interfaces so that they can be easily maintained.
- 2) Java package provides access protection.
- 3) Java package removes naming collision.

3. Explain JDK, JRE and JVM?

- 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 is platform dependent, JRE is also platform dependent, but JVM is platform independent.
- JDK contains tools for developing, debugging, etc. JRE contains class libraries and other supporting files, whereas software development tools are not included in JVM.
- JDK comes with the installer, on the other hand, JRE only contains the environment to execute source code whereas JVM is bundled in both software JDK and JRE.

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

- public- It is access specifier means from everywhere we can access it
- static - Access modifier means we can call this method directly using class name without creating an object of it
- void - It's the return type
- main - method name
- string [] args - in java accept only string type of argument and stores it in a string
- you can run the java program when you declared this method. program will start main method.

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

- C++ uses only compiler, whereas Java uses compiler and interpreter both.
- C++ supports both operator overloading & method overloading whereas Java only supports method overloading.
- 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 support structures.
- C++ supports unions while Java doesn't support unions.

6. Why Java is platform independent?

- In the case of Java, **it is the magic of Bytecode that makes it platform Independent.**
- This adds to an important feature in the JAVA language termed as **portability**. Every system has its own JVM which gets installed automatically when the jdk software is installed. For every operating system separate JVM is available which is capable to read the .class file or byte code.
- An important point to be noted is that while **JAVA is platform-independent language, the JVM is platform-dependent.** Different JVM is designed for different OS and byte code is able to run on different OS

7. What are wrapper classes in Java?

A Wrapper class is a class which contains the primitive data types (int, char, short, byte, etc). In other words, wrapper classes provide a way to use primitive data types (int, char, short, byte, etc) as objects. These wrapper classes come under java.util package.

- Wrapper Class will convert primitive data types into objects. The objects are necessary if we wish to modify the arguments passed into the method (because primitive types are passed by value).
- The classes in java.util package handles only objects and hence wrapper classes help in this case also.
- Data structures in the Collection framework such as ArrayList and Vector store only the objects (reference types) and not the primitive types.
- The object is needed to support synchronization in multithreading.

8. Why pointers are not used in Java?

- Java does have things that are pointers. But the terminology is different and you cannot use them freely like in C.
- In Java, types are divided into primitive types (int, long, short, char, byte, double, float, boolean) and reference types (everything else). Object,

String, Object[], etc. are all reference types. A value of a reference type is called a reference. A reference is a pointer to an object, unless it is the null reference.

9. List some features of Java?

Object Oriented

In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

Platform Independent

Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform-independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

Simple

Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

Secure

With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

Architecture Neutral

Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

Portable

Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. The compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

Robust

Java makes an effort to eliminate error-prone situations by emphasizing mainly on compile time error checking and runtime checking.

Multi Threaded

With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.

Interpreted

Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.

High Performance

With the use of Just-In-Time compilers, Java enables high performance.

Distributed

Java is designed for the distributed environment of the internet.

Dynamic

Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects at run-time.

10. Why is Java Architectural Neutral?

Java is architecture neutral because there are no implementation dependent features, for example, the size of primitive types is fixed.

In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. However, it occupies 4 bytes of memory for both 32 and 64-bit architectures in Java

11.How Java enabled High Performance?

Java is platform independent language, so it will be in any operating system . java has own library to build the programming. another most important thing which help to high performance is JIT & JVM. These two are very important to run the program.

and java has so many feature which is help to high performace. like oop, multithreading,Exception handling etc.

if we will talk about only for performance then the multithredaing has the biggest role.

12.Why Java is considered dynamic?

Java is considered dynamic because of Bytecode. The source code which is written in one platform that code can be executed in any platform. It loads the class file during runtime only. Hence, any thing that happens in runtime is dynamic.

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. As the name implies, the JVM acts as a "virtual" machine or processor. Java's platform independence consists mostly of its Java Virtual Machine (JVM) . JVM makes this possible because it is aware of the specific instruction lengths and other particularities of the platform (Operating System).

The JVM is not platform independent. Java Virtual Machine (JVM) provides the environment to execute the java file(. Class file). So at the end it's depends on your kernel , and kernel is differ from OS (Operating System) to OS. The JVM is used to both translate the bytecode into the machine

language for a particular computer, and actually execute the corresponding machine-language instructions as well. Without the JVM, you can't run a Java application.

14.List two Java IDE's?

Java has many IDE's that you can use. These IDE's or Integrated Development Environment provide immense help in the application development process. By using them, you can save time and effort as well as create a standard development process in your team or company. While the most popular Java IDE's in the world are Eclipse, IntelliJ IDEA, NetBeans, etc. there are many other IDE's that you can use according to your specifications.

Eclipse is a Java IDE that is one of the 3 biggest and most popular IDE's in the world. It was written mostly in Java but it can also be used to develop applications in other programming languages apart from Java using plug-ins

IntelliJ IDEA is a Java IDE that is one of the 3 biggest and most popular IDE's in the world. It has two versions, namely the free open-source community edition and a paid Ultimate edition.

15.Why Java is called as “Platform”?

The platform is a hardware or software used to run an application. The term Platform Independence means "write once and run anywhere". When we compile our Java code then .class file is generated by compiler these codes are readable by JVM and every operating system have its own JVM so JVM is platform dependent, due to JVM java language is platform independent.

16.Is Java Pure-Object oriented Language?

No, Java is not pure object oriented.

Object oriented programming language should only have objects whereas java have int,char,float which are not objects.

As in C++ and some other object-oriented languages, variables of Java's primitive data types are not objects. Values of primitive types are either stored directly in fields (for objects) or on the stacks (for methods) rather than on the heap, as is commonly true for objects. This was a conscious decision by Java's designers for performance reasons. Because of this, Java was not considered to be a pure object-oriented programming language.

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

We have learned Java 8 version.

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,
- Concurrency Enhancements,
- JDBC Enhancements etc.

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

In Java, the program is not converted to code directly understood by Hardware, rather it is converted to bytecode(.class file)which is interpreted by JVM, so once compiled it generates bytecode file, which can be run

anywhere (any machine) which has JVM(Java Virtual Machine) and hence it gets the nature of Write Once and Run Anywhere.

19.Difference between path and classpath.

Path variable is used to set the path for all Java software tools like javac.exe, java.exe, javadoc.exe, and so on whereas classpath variable is used to set the path for java classes.

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

The main() is the starting point for JVM to start execution of a Java program. Without the main() method, JVM will not execute the program. The syntax of the main() method is:

```
public static void main(String args []){ }
```

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.

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.

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.

1) Class Loader

The class loader is a subsystem used for loading class files. It performs three major functions viz. Loading, Linking, and Initialization.

2) Method Area

JVM Method Area stores class structures like metadata, the constant runtime pool, and the code for methods.

3) Heap

All the objects, their related instance variables, and arrays are stored in the heap. This memory is common and shared across multiple threads.

4) JVM language Stacks

Java language Stacks store local variables, and it's partial results. Each thread has its own JVM stack, created simultaneously as the thread is created. A new frame is created whenever a method is invoked, and it is deleted when method invocation process is complete.

5) PC Registers

PC register store the address of the Java virtual machine instruction which is currently executing. In Java, each thread has its separate PC register.

6) Native Method Stacks

Native method stacks hold the instruction of native code depends on the native library. It is written in another language instead of Java.

7) Execution Engine

It is a type of software used to test hardware, software, or complete systems. The test execution engine never carries any information about the tested product.

8) Native Method interface

The Native Method Interface is a programming framework. It allows Java code which is running in a JVM to call by libraries and native applications.

9) Native Method Libraries

Native Libraries is a collection of the Native Libraries (C, C++) which are needed by the Execution Engine.

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

Java compiler compiles source files (.java) to bytecode files (.class). Sun gives a free java compiler which is invoked with the 'javac' command.

A java interpreter is usually referred to as the Java Virtual Machine (or JVM). It reads and executes the bytecodes in the .class files. Java also supplies a free version of the JVM which is invoked with the 'java' command.

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

Yes. An empty .java file is a perfectly valid source file. Java file contains more than one **java** classes, provided at the most one of them is a public class.

27.Is JRE different for different Platforms ?

No . Each operating system has its own jre because its a platform dependent. java is platform independent means we wont need to change our code to run on different platform however we have to use platform specific jre versions to run the program.So different OS different JRE.

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

In Java, the Java Virtual Machine (JVM) that executes Java code has to log all objects being created (or references to them to be exact) so that the memory allocated for them can later be freed automatically by garbage collection when objects are not referenced any more.

In C++, only objects created on the stack are released automatically (when they get out of scope) unless you use some mechanism that handles this for you.

The main difference is that the JVM first initializes all members to zero, before starting to execute any constructor. In C++, member initialization is part of the constructor.

29.Who invokes main() function ?

The main method in the Java language is similar to the main function in C and C++. When the Java interpreter executes an application (by being invoked upon the application's controlling class), it starts by calling the class's main method. The main method then calls all the other methods required to run your application.

If you try to invoke the Java interpreter on a class that does not have a main method, the interpreter refuses to compile your program and displays an error message similar to this:

In class NoMain: void main(String argv[]) is not defined

30.What is .class file known as ?

.class file is known as Bytecode. JVM executes the bytecode and produces Operating System specific machine instructions.

**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 key component of the OpenJ9 VM that improves the performance of **Java** applications by **compiling** platform-neutral **Java** bytecode into native machine code at run time. Without the **JIT**, the VM has to interpret the bytecodes itself - a process that requires extra CPU and memory.

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

The **memory in the JVM** divided into **5 different parts**:

1. Class(Method) Area
2. Heap
3. Stack
4. Program Counter Register
5. Native Method Stack

Class Loader:It is a subsystem of JVM which is used to load class files.It is mainly responsible for three activities.

1. Loading
2. Linking
3. Initialization
- 4.

Class(Method) Area: It stores class level data of every class such as the runtime constant pool, field and method data, the code for methods.

Heap: It is used to allocate memory to objects at run time

Stack:

Each thread has a private JVM stack, created at the same time as thread. It is used to store data and partial results which will be needed while returning value for method and performing dynamic linking.

Java Stack stores frames and a new frame is created each time at every invocation of the method.

A frame is destroyed when its method invocation completes

Program Counter Register: Each JVM thread which carries out the task of a specific method has a program counter register associated with it. The non-native method has a PC which stores the address of the available JVM instruction whereas, in a native method, the value of the program counter is undefined. PC register is capable of storing the return address or a native pointer on some specific platform.

Native method Stacks: Also called as C stacks, native method stacks are not written in Java language. This memory is allocated for each thread when its created And it can be of a fixed or dynamic nature.

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

- Local members cannot use any of the access level since their scope is only inside the method.
- **Final** is the Only Non Access Modifiers 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?**1. Public**

The members, methods and classes that are declared public can be accessed from anywhere. This modifier doesn't put any restriction on the access.

2. Private

The scope of private modifier is limited to the class only.

- Private Data members and methods are only accessible within the class
- Class and Interface cannot be declared as private

- If a class has private constructor then you cannot create the object of that class from outside of the class.

3. Protected

Protected data member and method are only accessible by the classes of the same package and the subclasses present in any package. You can also say that the protected access modifier is similar to default access modifier with one exception that it has visibility in sub classes. Classes cannot be declared protected. This access modifier is generally used in a parent child relationship.

4. Default

When we do not mention any access modifier, it is called default access modifier. The scope of this modifier is limited to the package only. This means that if we have a class with the default access modifier in a package, only those classes that are in this package can access this class. No other class outside this package can access this class. Similarly, if we have a default method or data member in a class, it would not be visible in the class of another package.

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 referring 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 referring 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).

