Page No.

Date / /

Assignment :-02

Q1) what is difference between JDK, JRE, & JVM?

→ (i) Jdk stands for Java developement kit. It is used to develop Java application. the Jdk includes a compiler a debugger and other developement tools needed to create Java Programs in addition, it includes the JRE (Java runtime Envoirnment) which is needed to run Java application.

(ii) JRE stands for Java runtime Envoirnment it is a software Package that Provides the minimum requirement for executing a Java application. The JRE includes the JVM (Java virtual machine) the class libraries and other components needed to run Java Program.

(iii) JVM stands for Java virtual machine. It is an abstract machine that Provides a runtime envoirnment in which Java bytecode can be Executed. the JVM interPrets the bytecode and translate it into machine-specific instruction. the Jvm is an important part of the Java Platform because it Povide Platform independance, allowing Java Programs to run on any platform that has a compatible Jvm installed.

Q2]

Shrikrupa

Page No.
Date / /

q2] what is JIT compiler ?

→ JIT (Just in time) compiler is a type of compiler that is used by the Java virtual machine (JVM) to improve the performance of Java application. the JIT compiler is part of the runtime environment of the JVM.

when a Java application is executed. the Java virtual machine loads the bytecode of the application into memory and interprets it interpreting the bytecode can be slower than executing compiled code at runtime the JIT compiler improves the performance of the application by compiling frequently executed sections of bytecode into machine code at runtime.

It is the one of the key features of the Java platform that allows Java application to run with good performance without JIT compiler, the performance of Java applications would be significantly slower, especially for long-running applications.

Date __/__/__

93]. What is class loader?

→ in Java, a class loader is a part of the Java runtime Envoienment (JRE) that is responsible for loading Java class into memory at runtime. when a java application is executed the class loader loads the classes needed by the application into memory and makes them available for use. the class loader is responsible for finding and loading the bytecode of a class from the file system or from the other sources. which are network socket, database or even other java archieves it then convert the bytecode into a java class object and load it into memory the class loader also resolves any dependancies in that class may behave on other class & load this class as well.

their are 3 main types of class loaders in java,

1) Bootstrap class loader.
2) Extension class loader.
3) Application class loader.

Date / /

94) Explain various memory logical partitions?

→ The main memory partitions in java are.

1. Heap memory :- this is memory area used to store and create Java objects. and it is managed by JVM. and divided into 2 region the young generation & old generation. the young generation is further divided into Eden space & survivor space. new objects are initially allocated in the Eden space. and then moved to the survivor space for further processing. when an object survives a number of garbage collections cycles, it is promoted to the old generation.

2. ~~stack memory~~ method area :- this is the memory area where the class metadata, static variables and method codes are stored. the method area is shared among all threads in a java applications and is managed by the JVM.

3. stack memory :- this is the memory are where the call stack of a thread is stored. each thread in java has its own stack memory which is used to store local variable, method parameters and other data related to the execution of the thread. the stack memory is managed by the jvm & is automatically allocated & deallocated as needed.

Page No.

Date / /

4. Native heap :- this is the memory area where native libraries and data is stored. Native libraries are loaded by the JVM into this area of memory and are used by the Java application to interact with the underlying O.S.

5. Direct memory :- This is the memory area that is used for off-heap memory management direct memory is allocated outside of the JVM heap and is used for storing large amounts of data that need to be accessed quickly.

Page No.
Date / /

Q5) What gives Java its "write once and run anywhere nature"?

→ Java's "write once and run anywhere" nature is acheived through the use of the Java virtual machine (JVM) & the platform's architecture.

When a Java program is compiled it is compiled into bytecode, which is a platform independant representation of the program. this bytecode can be executed on any platform that has a JVM installed, without the need for recompilation. the JVM is responsible for translating the bytecode into platform-specific machine code at runtime allowing the program to run on any platform that supports the JVM

this approach allows Java program to be written once and executed on any platform that has a JVM, without the need for recompilation or modification of the program's source code.

Page No.
Date / /

Q6) Explain history of Java. who invented Java?

→ Java was created by Jame gosling, mike sheridan, and patrick naughton at sun microsystem (which later acquired by oracle corporation) the development of java started from early 1990s, with the team aiming to create a langauge that could be used to build platform-independant application for consumer electronics devices.

the first version of Java called oak, was released in 1995 the name "Java was choosen later. that year after the team discovered that name oak was already treadmark by another company.

Java quickly became popular among developers, and its use spread rapidly across a wide range of industries and application including enterprise software development, web development, and mobile app development.

Date / /

98) list features of Java. ?

→ Java is a powerful & versatile Programming lang. that offers a wide range of features and capabilities

1. Platform independance :- Java code can be Run on any platform that has a java virtual machine (JVM) installed which makes it highly portable & adaptable to different operating system & hardware.
2. OOP :- Java is object oriented Programming language. which means that it supports the creation of modular reusuable code that is easy to maintain & extend.
3. Automatic memory management :- Java's automatic memory management system (known as memory mangement system) garbage collection. simplifies memory management & reduces the risk of memory leaks & other common programming errors.
4. Robustness and reliability :- Java's strict type checking, exception handling & other built in features crashes.
5. multithreading :- Java supports multithreading which allows for the creation of concurrent and parallel programs that can take advantage of modern hardware architectures.
6. security :- it includes range of security features that help Protect against viruses, malware & other security threats, includings sandboxing, digital signature and encryption
7. Rich standard library :- Java includes rich set of standard libraries & APIs that Provide developes wide range of tools & functionality for building Applications.

Shrikrupa

Date __/__/__

99) List various Datatypes in Java ?

→ Java support Wide Range of data types.

1. byte - 8-bit signed integer (Range → -128 to 127)
2. short - 16 bit signed integer (-32768 to 32767)
3. int - 32 bit signed integer (-2^31 to 2^31 - 1)
4. long - 64 bit signed integer (-2^63 to -2^63 - 1)
5. float - 32 bit floating point number.
6. double - 64 bit floating point number.
7. boolean - true / false value.
8. char - 16 bit unicode character.

some additional object data type.

1. string :- a sequence of characters.
2. Array :- a collection of element of same data type.
3. class :- an object that represent a java class
4. interface :- a type that defines a set of method that a class must implement.
5. Enum :- a type that defines a fixed set of constant value.

Page No.
Date / /

Q10) what is difference between. system.out.print system.out.println system.err.print.

→ system.out.print() is used to print output without a newline character.

system.out.println() is used to print output with a newline character.

system.err.print() is used to print error message to the console.

Q11) How the Java platform is independent ?

→ 1) bytecode ⇒ Java source code is compiled into platform-independent bytecode which is a highly optimized & compact code that can be executed on any platform that has JVM installed.

2) JVM :- it is an interpreter which can execute Java code in any machine which has JVM installed. without any charges this makes it portable & adaptable

3) standard libraries :- this allowing developer to write platform independent code this libraries provide a wide range of tools & functionality that developer can be used to build application.

Shrikrupa

Date __/__/__

Q12) what is bytecode? how it is different from machine code?

→ bytecode is an intermediate represnt-ation of a program's source code that is generated by the Java compiler it is a binary format that is designed to be executed on a Java virtual machine (JVM). the bytecode is highly optimized & compact code that is platform independent, meaning that it can be executed on any platform that has a JVM installed.

bytecode is different from machine code in the following ways-

1) platform independance
2) interpretation.
3) compactness.

Date / /

Q13) what is difference bet" Jar file & runnable Jar file?

→ The main difference between a Jar file & Runnable Jar file is that the letter has been marked as executable & contains a main method that can be used to start the application. A Jar file on the other hand may or may not contain a main method & it is typically used as a library or a collection of Related resource.

Q14) what is difference between Runnable Jar file & exe file?

→ both used to distribute & run application, but they are different in several way.

1. Platform :- A Runnable Jar file is a platform-independant format that can run on way platform that has a Java virtual machine (JVM) installed. An exe file on the other hand, is specific to the window platform & can be only Run on windows based system.
2. Execution :- Jar file reads bytecode & interprets it into machine code that can be executed by the underlying hardware. Exe. file is binary file that is directly executable by hardware
3. Packging :- A Runnable JAR is packaged as a single archeive file that contains all the nessesary resources and code to run the application. an exe file requires additional installation files & resources to be distributed & installed properly.

Date / /

Q15) How is C platform dependent langauge?

→

C is a platform-dependent langauge because it is compiled into machine code that is specific to a particular computer architecture & operating system. the machine code generated by the C compiler is designed to be executed on a specific platform, which means that it may not be compatible with other platforms.

C programs are written using platform-specific libraries & system calls which are used to interacts with the underlying hardware & operating system. these libraries and system calls may have different implementation & behaviors on different platforms which can lead to portability issues when moving a C program from one platform to another.

to make a C program portable across multiple platform it may be necessary to modify the code to use platform-independent libraries & techniques or to write platform-specific code for each target platform. this can be a complex & time-consuming process which is why C is often considered a platform-dependent langauge.

Date __/__/__

9) what is difference between Path & class path?

1. Path :- Path refers to the system variable that specifies the location of executable files and other resources on the operating system the system uses the path variable to locate executable files when a user types a command at the command prompt in Java the path variable is used to locate the Java compiler & other tools that are required to build and run Java Programs.

2. classPath :- classPath is an envoironment variable that specifies the location of Java class files & other resources that are required by a Java Program. when a Java Program is run the Java virtual machine (JVM) uses the classPath to locate the class files that are required to run the Program.