Selvam College of Technology

An Autonomous Institution

Accredited by NAAC with "A" Grade, UGC Recognized 2(f) Status,
An ISO 9001:2015 Certified Institution, Approved by AICTE New Delhi, Affiliated to Anna University-Chennai
PONNUSAMY NAGAR, SALEM ROAD(NH-44), NAMAKKAL-637003. TAMILNADU.
Mobile: 9942099122, 9942099109, Web: www.selvamtech.edu.in

Name of the Bundle	Advanced Bundle V1	**Subject**	Java Programming V1
Topic	Introduction to Java	**Last updated on**	07 August 2024

1. What is java?

   a. A physical machine

   b. A software tool

   c. A programming language and a platform

   d. A tool which creates another software

Ans: c. A programming language and a platform

Explanation: As a platform, it provides a runtime environment that enables Java applications to run on various devices and operating systems without modification.

2. Who is the founder of Java?

   a. Dennis Ritchie

   b. James Gosling

   c. Rasmus Lerdorf

   d. Brendan Eich

Ans: b. James Gosling

Explanation: James Gosling is known as the "father of Java" for his role in creating the Java programming language.

3. When was the Java language project initiated?

   a. June 1991

   b. April 1995

   c. Jan 1996

   d. Feb 1997

Ans: a. June 1991

Explanation: The Java language project was initiated in June 1991.

Name of the Bundle	Advanced Bundle V1	**Subject**	Java Programming V1
Topic	Introduction to Java	**Last updated on**	07 August 2024

4. Initially Java was called ________.

  a. Java
  b. Green
  c. Oak
  d. Green-talk

Ans: c. Oak

Explanation: Initially, Java was called "Oak." The name was later changed to Java. Because it is a type of coffee, and the team wanted a name that was fun and associated with energy.

5. What kind of Language is Java?

  a. Procedural
  b. Non-Procedural
  c. Object Oriented
  d. Event Driven

Ans: c. Object Oriented

Explanation: Java is an object-oriented programming language. It emphasizes objects and classes, promoting concepts like encapsulation, inheritance, and polymorphism.

6. Which type of applications are created through the Java Programming Language?

  a. Standalone Application
  b. Web Application
  c. Enterprise Application & Mobile Application
  d. All of the above

Ans: d. All of the above

Explanation: Java is versatile and can be used to create various types of applications, including standalone applications, web applications, enterprise applications, and mobile applications.

7. Standalone Applications are known as ______.

   a. Distributed applications
   b. Server-side applications or Web applications
   c. Desktop applications or Window-based applications
   d. Android applications or Java ME applications

Ans: c. Desktop applications or Window-based applications

Explanation: Standalone applications are commonly known as desktop applications or window-based applications.

8. Web Applications are known as_____.

   a. Distributed applications
   b. Server-side applications or Web applications
   c. Desktop applications or Window-based applications
   d. Android applications or Java ME applications

Ans: b. Server-side applications or Web applications

Explanation: They are designed to run on web servers and can be accessed through a web browser.

9. Enterprise Applications are known as______.

   a. Distributed applications
   b. Server-side applications or Web applications
   c. Desktop applications or Window-based applications
   d. Android applications or Java ME applications

Ans: a. Distributed applications

Explanation: They are designed to handle large-scale, complex business processes and typically involve multiple interconnected systems and components.

10. Mobile Applications are known as _______.

a. Distributed applications

b. Server-side applications or Web applications

c. Desktop applications or Window-based applications

d. Android applications or Java ME applications

Ans: d. Android applications or Java ME applications

Explanation: Mobile applications are known as Android applications or Java ME (Micro Edition) applications. Java ME was used for developing applications for feature phones, while Android applications are specifically for devices running the Android operating system.

11.Which of the following is not a Java feature?

a. Dynamic

b. Architecture Neutral

c. Use of pointers

d. Object-oriented

Ans: c. Use of pointers

Explanation: Java does not use pointers. Instead, it provides a safer alternative through references. The features of Java include being dynamic, architecture-neutral, and object-oriented.

12. Which of the following options leads to the portability and security of Java?

a. Bytecode is executed by JVM
b. The applet makes the Java code secure and portable
c. Use of exception handling
d. Dynamic binding between objects

Ans: a. Bytecode is executed by JVM

Explanation: Bytecode allows Java applications to run on any device or operating system with a compatible JVM, while the JVM provides a layer of security by managing memory and enforcing access controls.

13.Which feature of Java makes it possible to run a Java program on different platforms?

a. Object-Oriented
b. Platform-Independent
c. Syntax
d. Memory Management

Ans: b. Platform-Independent

Explanation: This is achieved through Java's use of bytecode, which can be executed on any platform with a compatible Java Virtual Machine (JVM).

14. The command javac is used to __________.

a. debug a java program
b. compile a java program
c. Interpret a java program
d. Execute a java program

Ans: b. compile a java program

Explanation: The java c command is used to compile a Java program. It translates the Java source code into bytecode, which can then be executed by the Java Virtual Machine (JVM).

15. What kind of language is Java considered to be?

a. Compiled

b. Interpreted

c. Both compiled and interpreted

d. None of the above

Ans: c. Both compiled and interpreted

Explanation: Java is considered to be both compiled and interpreted. The Java source code is first compiled into bytecode using the javac compiler. This bytecode is then interpreted or executed by the Java Virtual Machine (JVM), which can also perform Just-In-Time (JIT) compilation to improve performance.

16. Java programs are compiled into ___________.

a. Assembly language code

b. Machine code

c. Bytecode

d. Source code

Ans: c. Bytecode

Explanation: This bytecode is platform-independent and can be executed by the Java Virtual Machine (JVM) on any device or operating system.

17. In Java, bytecode is _______________.

a. The code generated by the Java interpreter from source code.

b. The native code generated by the JVM during execution.

c. A type of intermediate code that can be executed by the Java Virtual Machine (JVM).

d. The final executable file after compilation.

Ans: c.A type of intermediate code that can be executed by the Java Virtual Machine (JVM).

Explanation: It is generated by the Java compiler from the source code and allows Java programs to be platform-independent.

18. Which component of Java is responsible for interpreting bytecode into machine code?

a. Java Compiler
b. Java Virtual Machine (JVM)
c. Java Runtime Environment (JRE)
d. Java Development Kit (JDK)

Ans: b. Java Virtual Machine (JVM)

Explanation: The Java Virtual Machine (JVM) is responsible for interpreting bytecode and converting it into machine code that can be executed by the host system.

19. What is the primary purpose of the Java Virtual Machine (JVM)?

a. To compile Java source code into bytecode.

b. To interpret Java bytecode into machine code.

c. To execute Java bytecode on different platforms.

d. To provide an IDE for Java development.

Ans: c. To execute Java bytecode on different platforms.

Explanation: This allows Java programs to be platform-independent by running on any system with a compatible JVM.

20.Which component of the JVM is responsible for converting bytecode to machine code?

a. Class Loader
b. JIT Compiler
c. Bytecode Verifier
d. Interpreter

Ans: d. Interpreter

Explanation: The Interpreter in the JVM is responsible for converting bytecode to machine code. It reads the bytecode line by line and translates it into executable machine code for the underlying hardware. This process allows Java programs to run on different platforms.

21. Which memory area is used by the JVM to store class structures and method data?

a. Heap
b. Class (Method) Area
c. Stack
d. Native Method Stack

Ans: b. Class (Method) Area

Explanation: The Class (Method) Area is used by the JVM to store class structures, method data, and runtime constant pools. It is part of the memory area allocated for the JVM's internal use.

22. Which of the following is NOT a component of the JVM?

a. Java Compiler
b. JIT Compiler
c. Bytecode Generator
d. Class Loader

Ans: c. Bytecode Generator

Explanation: The Bytecode Generator is part of the Java compiler. The Java compiler (javac) compiles Java source code into byte code.

23. The JVM specification ensures _______.

a. Platform independence for Java programs.
b. Efficient compilation of Java source code.
c. Compatibility with all programming languages.
d. High-level security for Java applications.

Ans: a. Platform independence for Java programs.

Explanation: This means that Java bytecode can be executed on any platform that has a compatible JVM, allowing Java applications to run on different operating systems without modification.

24.Which JVM component manages memory allocation for Java objects during runtime?

a. memory manager
b. JIT Compiler
c. Garbage Collector
d. Bytecode Verifier

Ans: a. memory manager

Explanation: When a new object is created, the JVM allocates memory for that object in the heap space.

25. Which memory area of the JVM stores local variables and partial results?

a. Heap
b. Method Area
c. Stack
d. Native Method Stack

Ans: c. Stack

Explanation: Stores local variables, method parameters, and partial results for each method call, crucial for managing method execution.

26. JDK stands for____________.

a. Java Development Kit

b. Java Virtual Machine

c. Java Run-time Environment

d. None of Above

Ans: a. Java Development Kit

Explanation: Stands for Java Development Kit, providing tools like the Java compiler and debugger for developing Java applications.

27. What is the primary function of the Java Development Kit (JDK)?

a. To compile Java source code into bytecode.

b. To interpret Java bytecode into machine code.

c. To provide tools for developing and debugging Java applications.

d. To execute Java bytecode on different platforms.

Ans: c. To provide tools for developing and debugging Java applications.

Explanation: Supplies essential tools for Java development, such as the compiler for converting code into bytecode and debugging tools for identifying and fixing issues.

28.Which component of the JDK is responsible for compiling Java source code?

a. JRE (Java Runtime Environment)

b. JVM (Java Virtual Machine)

c. javac (Java Compiler)

d. JDK Compiler

Ans: c. javac (Java Compiler)

Explanation: The javac (Java Compiler) is responsible for compiling Java source code into bytecode, which is then executed by the JVM. It is a key component of the JDK, converting .java files into .class files.

29. Which of the following is a component of JVM.

a. Class loader
b. Java Interpreter
c. JIT Compiler
d. All the above

Ans: d. All the above

Explanation: The components of the Java Virtual Machine (JVM) include the Java Runtime Environment, Java Interpreter, and Just-In-Time (JIT) Compiler.

30. Which tool from the JDK is used to create executable JAR files?

a. javac
b. java
c. javap
d. jar

Ans: d. jar

Explanation: The jar tool is used to package Java classes and resources into a single JAR (Java Archive) file that can be executed.

31. Which component of the JDK is responsible for running Java applications?

a. JRE (Java Runtime Environment)
b. JVM (Java Virtual Machine)
c. javac (Java Compiler)
d. JDK Compiler

Ans: a. JRE (Java Runtime Environment)

Explanation: The JRE (Java Runtime Environment) provides the JVM and libraries needed to run Java applications.

32.Which command is used to check the version of the JDK installed on your system?

a. javac -version
b. java -version
c. jdk -version
d. version -jdk

Ans: b. java –version

Explanation: The java -version command shows the version of the Java Runtime Environment (JRE) and, indirectly, the JDK version installed on your system.

33. Which statement best describes the Java Virtual Machine (JVM)?

a. A physical machine used to execute Java applications.
b. A software-based machine that executes Java bytecode.
c. A hardware component designed for Java compilation.
d. An integrated development environment for Java programming.

Ans: b. A software-based machine that executes Java bytecode

Explanation: Best description of the Java Virtual Machine (JVM): The JVM is a software-based machine that executes Java bytecode, allowing Java programs to run on any platform.

34. The JVM is an example of?

a. A hypervisor-based virtual machine.
b. An application-level virtual machine.
c. A kernel-based virtual machine.
d. A hardware-based virtual machine.

Ans: b. An application-level virtual machine.

Explanation: The JVM is an application-level virtual machine because it provides a runtime environment for running Java applications on top of an operating system.

35. JRE stands for?

a. Java Run-time Kit

b. Java Readable Machine

c. Java Run-time Environment

d. None of Above

Ans: c. Java Run-time Environment

Explanation: It provides the necessary environment to run Java applications.

36. The JRE is an essential component for ________________.

a. Compiling Java source code.

b. Debugging Java applications.

c. Executing Java applications.

d. Writing Java documentation.

Ans: c. Executing Java applications.

Explanation: It provides the necessary runtime environment for running Java programs.

37. Which JDK tool is NOT part of the JRE?

a. javac

b. java

c. javap

d. jar

Ans: a. Javac

Explanation: javac is a Java compiler used for compiling source code into bytecode and is part of the JDK, not the JRE.

38. What is the primary role of the Just-In-Time (JIT) compiler in Java?

a. To compile Java source code into bytecode.

b. To interpret Java bytecode into machine code.

c. To optimize and compile Java bytecode into native machine code.

d. To manage memory allocation for Java objects.

Ans: c. To optimize and compile Java bytecode into native machine code.

Explanation: The JIT compiler's role is to optimize and compile Java bytecode into native machine code, improving runtime performance by converting bytecode into machine code that runs directly on the hardware.

39. The JIT compiler in Java is part of which JVM component?

a. Bytecode Interpreter

b. Class Loader

c. Garbage Collector

d. Execution Engine

Ans: d. Execution Engine

Explanation: The JIT compiler is part of the Execution Engine, which is responsible for running and optimizing bytecode execution.

40. Which type of optimization is typically performed by the JIT compiler?

a. Early binding

b. Static binding

c. Runtime binding

d. Method inlining

Ans: d. Method inlining

Explanation: Method inlining is a common optimization where the compiler replaces a method call with the method's code to reduce.

41. When does JIT compilation occur in relation to Java bytecode execution?

a. Before bytecode execution starts.
b. During bytecode interpretation.
c. After bytecode execution completes.
d. Just before JVM initialization.

Ans: b. During bytecode interpretation.

Explanation: During bytecode interpretation, the JIT compiler optimizes and compiles bytecode into native machine code to improve performance.

42. Which of the following is the correct sequence of steps in the execution of a Java program?

a. Compile→ Load→ Execute
b. Load→ Compile→ Execute
c. Execute→ Load→ Compile
d. Compile → Execute → Load

Ans: a. Compile→ Load→ Execute

Explanation: Compile → Load → Execute. First, the Java source code is compiled into bytecode, then the bytecode is loaded into the JVM, and finally, it is executed.

43. After compiling a Java program, what type of file is generated?

a. .class file
b. .java file
c. .exe file
d. .obj file

Ans: a. .class file

Explanation: The compiler generates a .class file, which contains the bytecode that the JVM can execute.

44. Which method is the entry point of any Java program?

a. init ()
b. start ()
c. main ()
d. run ()

Ans: c. main ()

Explanation: The main () method is the entry point where the Java application starts execution.

45. What is the extension of the Java bytecode file?

a. .java
b. .exe
c. .class
d. .byte

Ans: c. .class

Explanation: The bytecode file has a .class extension, which contains the compiled bytecode of a Java program.

46. Which of the following is a benefit of using JIT compilation?

a. Reduced startup time
b. Better runtime performance
c. Simplified source code
d. Lower memory consumption

Ans: b. Better runtime performance

Explanation: Better runtime performance. JIT compilation improves performance by converting bytecode to native code during execution, which runs faster than interpreting bytecode.