Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

1. What is a package in Java?
    a. A mechanism to organize related classes, interfaces, and sub-packages.
    b. A data structure to store variables.
    c. A type of method in Java.
    d. A keyword for inheritance.

Ans: a. A mechanism to organize related classes, interfaces, and sub-packages.

Explanation: A package in Java is used to group related classes and interfaces together to avoid naming conflicts and improve organization.

2. How do you declare a package in Java?
    a. package;
    b. package packageName;
    c. package = packageName;
    d. class packageName;

Ans: b. package packageName;

Explanation: The package keyword is used to declare the package, followed by the name of the package.

3. Which of the following is NOT a reason to use packages in Java?
    a. To avoid name conflicts between classes.
    b. To make the code more organized.
    c. To allow classes to be imported from other packages.
    d. To make all classes in a package private.

Ans: d. To make all classes in a package private.

Explanation: Packages help organize code and prevent naming conflicts. They don't automatically make classes private; access control is achieved using modifiers like private, protected, and public.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

4. What is the default package in Java if no package is specified?
    a. java.package
    b. package java;
    c. default
    d. No package is assigned.

Ans: d. No package is assigned.

Explanation: If no package is explicitly declared, the class is considered to be in the default package.

5. How do you import a class from another package in Java?
    a. import packageName.className;
    b. import className from packageName;
    c. use packageName.className;
    d. include packageName.className;

Ans: a. import packageName.className;

Explanation: The import statement is used to import a class from a specific package.

6. Why are packages helpful in Java?
    a. They make Java programs slower.
    b. They help organize Java code and improve readability.
    c. They make it difficult to access classes.
    d. They eliminate the need for inheritance.

Ans: b. They help organize Java code and improve readability.

Explanation: Packages make the code more organized and easier to maintain.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

7. What is one of the main reasons for using packages in Java?
    a. To prevent inheritance.
    b. To prevent naming conflicts between classes.
    c. To limit the number of classes in a program.
    d. To increase the complexity of the program.

Ans: b. To prevent naming conflicts between classes.

Explanation: By organizing classes in different packages, Java prevents classes with the same name from conflicting.

8. How do packages make searching and locating easier?
    a. By allowing classes to have random names.
    b. By organizing related classes and interfaces together.
    c. By removing all class methods.
    d. By making all classes public.

Ans: b. By organizing related classes and interfaces together.

Explanation: Packages group related classes, making it easier to find them.

9. What types of Java elements can be grouped within a package?
    a. Only classes
    b. Only methods and variables
    c. Classes, interfaces, enumerations, and annotations
    d. Only interfaces

Ans: c. Classes, interfaces, enumerations, and annotations

Explanation: A package can contain all Java elements including classes, interfaces, enumerations, and annotations.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

10. How does a package help in providing controlled access?
    a. By making all classes public.
    b. By restricting access to only specific classes and members within a package.
    c. By hiding all classes and methods.
    d. By making all methods private.

Ans: b. By restricting access to only specific classes and members within a package.

Explanation: Java packages allow access control using modifiers like public, protected, and private.

11. How does a package help in data encapsulation?
    a. By allowing only public access to class members.
    b. By grouping all data members in one class.
    c. By hiding implementation details and providing controlled access to data.
    d. By making data members accessible only to other packages.

Ans: c. By hiding implementation details and providing controlled access to data.

Explanation: Packages support data encapsulation by restricting access to class members using access control modifiers.

12. How do packages promote reusability of existing classes?
    a. By preventing inheritance.
    b. By making it difficult to import classes.
    c. By allowing existing classes to be used across different programs.
    d. By removing the need for interfaces.

Ans: c. By allowing existing classes to be used across different programs.

Explanation: Packages make it easier to reuse classes across different programs by simply importing them.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

13. How are packages categorized in Java?
    a. Built-in packages only
    b. User-defined packages only
    c. Built-in and user-defined packages
    d. None of the above

Ans: c. Built-in and user-defined packages

Explanation: Java has built-in packages (like java.util) and user-defined packages created by developers.

14. Which of the following is true about Java classes and packages?
    a. Every class is part of a certain package.
    b. Only public classes are part of a package.
    c. Only user-defined classes are part of packages.
    d. Every class must be part of a built-in package.

Ans: a. Every class is part of a certain package.

Explanation: Every class in Java belongs to a package, whether it's the default package or a custom package.

15. What are the two main types of packages in Java?
    a. Static and dynamic packages
    b. Built-in and user-defined packages
    c. Internal and external packages
    d. Public and private packages

Ans: b. Built-in and user-defined packages

Explanation: Java has built-in packages provided by the Java API, and user-defined packages created by the programmer.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

16. Which package type is provided by the Java platform?
    a. User-defined package
    b. Built-in package
    c. Internal package
    d. Private package

Ans: b. Built-in package

Explanation: Java provides built-in packages as part of the standard library, such as java.util.

17. What is the advantage of using a user-defined package in Java?
    a. To automatically import all Java classes.
    b. To organize and manage custom classes and interfaces.
    c. To reduce the complexity of built-in packages.
    d. To remove the need for inheritance.

Ans: b.To organize and manage custom classes and interfaces.

Explanation: User-defined packages allow developers to create their own organizational structure for custom classes and interfaces.

18. What is the purpose of the import statement in Java?
    a. To remove classes from a package
    b. To add user-defined packages to the program
    c. To define a new class
    d. To create a new package

Ans: b.To add user-defined packages to the program

Explanation: The import statement is used to include classes from other packages into the current program.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

19. How do you import a class from a user-defined package in Java?
    a. import packageName.className;
    b. import className from packageName;
    c. import packageName.ClasName;
    d. import className;

Ans: a. import packageName.className;

Explanation: The import statement specifies the full path (package name and class name) of the class to be used in the program.

20. Which of the following is a correct way to import a class from a subpackage in Java?
    a. import myPackage.MyClass;
    b. import myPackage.mySubPackage.MyClass;
    c. import MyClass from myPackage;
    d. import MyClass myPackage.mySubPackage;

Ans: b. import myPackage.mySubPackage.MyClass;

Explanation: To import a class from a subpackage, the full package path (including the subpackage) must be specified.

21. How can you access a class from a user-defined package myPackage?
    a. import myPackage.MyClass;
    b. import MyClass from myPackage;
    c. import MyClass;
    d. import myPackage.MyClass.mySubPackage;

Ans: a. import myPackage.MyClass;

Explanation: The class MyClass can be accessed by importing it from the myPackage package.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

22. What does import myPackage.mySubPackage.MyClass; do in Java?
    a. It defines a new class in myPackage.mySubPackage.
    b. It imports the MyClass from the mySubPackage within myPackage.
    c. It imports all classes from mySubPackage.
    d. It imports MyClass from the default package.

Ans: b. It imports the MyClass from the mySubPackage within myPackage.

Explanation: This statement imports the MyClass from a specific subpackage.

23. What does the statement import package.name.*; do in Java?
    a. Imports only the specified class from package.name.
    b. Imports all classes from package.name but not its subpackages.
    c. Imports all classes and subpackages from package.name.
    d. Imports the entire package.name as a new package.

Ans: b. Imports all classes from package.name but not its subpackages.

Explanation: The * wildcard imports all classes from the specified package, but it does not include subpackages.

24. What does the statement import java.util.Scanner; do in Java?
    a. Imports the entire java.util package.
    b. Imports the Scanner class from the java.util package.
    c. Imports all classes from the java.util package.
    d. Imports all subpackages of the java.util package.

Ans: b. Imports the Scanner class from the java.util package.

Explanation: This import statement includes the Scanner class from the java.util package for use in the program.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

25. What is the meaning of the statement import java.util.*; in Java?
    a. It imports only the Scanner class from java.util.
    b. It imports all classes from the java.util package.
    c. It imports all classes and subpackages from java.util.
    d. It imports only the java.util package but not the classes.

Ans: b. It imports all classes from the java.util package.

Explanation: The * wildcard imports all classes from the specified package.

26. What is the role of java.util in the statement import java.util.Scanner;?
    a. It is the class Scanner.
    b. It is the subpackage under java.util.
    c. It is the package containing the Scanner class.
    d. It is the entire java.util package.

Ans: c. It is the package containing the Scanner class.

Explanation: java.util is the package that contains the Scanner class.

27. How do you declare a package in Java?
    a. package mypack;
    b. package name = mypack;
    c. import mypack;
    d. class mypack;

Ans: a. package mypack;

Explanation: The package keyword is followed by the name of the package to declare it.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

28. What is the purpose of the package statement in Java?
    a. It imports the mypack package.
    b. It specifies that the source file belongs to the mypack package.
    c. It creates a new class named mypack.
    d. It allows classes from mypack to be used in other files.

Ans: b. It specifies that the source file belongs to the mypack package.

Explanation: The package statement tells Java that the current source file is part of the specified package.

29. Where should the package statement appear in a Java source file?
    a. At the bottom of the source file.
    b. Before any class declarations, as the first line of the source file.
    c. After the import statements.
    d. Inside the main method.

Ans: b. Before any class declarations, as the first line of the source file.

Explanation: The package statement should be the first line in the file before any class or import statements.

30. What happens if a Java source file doesn't include a package statement?
    a. It will be considered part of the default package.
    b. It will result in a compilation error.
    c. It will import all classes automatically.
    d. It will be unable to use any classes.

Ans: a. It will be considered part of the default package.

Explanation: If no package is specified, the class is considered to be in the default package.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

31. If you use package mypack;, what does that mean?
    a. The class is part of the mypack package.
    b. The class is imported from the mypack package.
    c. The class cannot be used in other packages.
    d. The class is stored in a mypack directory.

Ans: a. The class is part of the mypack package.

Explanation: This statement means that the class belongs to the mypack package.

32. What can be included inside a Java package?
    a. Only classes
    b. Classes, interfaces, enumerations, and subpackages
    c. Only classes and interfaces
    d. Only classes and enumerations

Ans: b. Classes, interfaces, enumerations, and subpackages

Explanation: A Java package can contain various elements like classes, interfaces, enumerations, and even subpackages.

33. Which of the following is NOT included inside a package in Java?
    a. Classes
    b. Interfaces
    c. Variables
    d. Subpackages

Ans: c. Variables

Explanation: Variables are not typically part of a package. Packages contain classes, interfaces, and other elements.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

34. What can a package contain besides classes in Java?
    a. Only methods
    b. Interfaces and enumerations
    c. Only variables
    d. Only subpackages

Ans: b. Interfaces and enumerations

Explanation: In addition to classes, packages can contain interfaces, enumerations, and annotations.

35. Can a Java package contain other packages?
    a. Yes, a package can contain subpackages.
    b. No, packages cannot contain subpackages.
    c. Yes, but only classes can be inside subpackages.
    d. No, a package can only contain classes.

Ans: a. Yes, a package can contain subpackages.

Explanation: A package can have subpackages, allowing a hierarchical structure.

36. What are subpackages in Java packages?
    a. Subclasses inside a package
    b. Additional packages contained within another package
    c. Special variables inside a package
    d. Interfaces inside a package

Ans: b. Additional packages contained within another package

Explanation: Subpackages are packages that reside inside another package, allowing for further categorization.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

37. How many package statements can there be in a single Java source file?
    a. None
    b. One
    c. Two
    d. Multiple

Ans: b. One

Explanation: Only one package statement is allowed per source file.

38. What happens if a Java source file doesn't have a package statement?
    a. It will cause a compilation error.
    b. It will be automatically assigned to the default package.
    c. It will be placed in the mypack package.
    d. It will be considered a public class.

Ans: b. It will be automatically assigned to the default package.

Explanation: If no package is declared, the class will belong to the default package.

39. What is the correct order of statements in a Java source file?
    a. Import statement, package statement, class definition
    b. Package statement, import statement, class definition
    c. Class definition, import statement, package statement
    d. Package statement, class definition, import statement

Ans: b. Package statement, import statement, class definition

Explanation: The package statement comes first, followed by imports, and then the class definition.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

40. What happens if a Java file has multiple package statements?
    a. It will compile and work as expected.
    b. Only the first package statement will be used, and others will be ignored.
    c. Compilation will fail with an error.
    d. It will create multiple packages for the same file.

Ans: c. Compilation will fail with an error.

Explanation: A Java file can have only one package statement, so multiple package statements will cause a compilation error.

41. Where should the package statement appear in a Java source file?
    a. After the class declarations
    b. Before any import statements
    c. After the import statements
    d. As the last line of the source file

Ans: b. Before any import statements

Explanation: The package statement must appear before any imports in the source file.

42. How is a package stored in Java?
    a. A package is stored as a .jar file.
    b. A package is stored in a folder with the same name as the package.
    c. A package is stored in the bin directory.
    d. A package is stored in the default src folder.

Ans: b. A package is stored in a folder with the same name as the package.

Explanation: Java packages are stored in directories or folders, and the folder name corresponds to the package name.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

43. What is the requirement for classes in a package that need to be accessed outside the package?

a. They must be declared private.
b. They must be declared public.
c. They must be declared static.
d. They must be declared final.

Ans: b. They must be declared public.

Explanation: To be accessed outside of the package, classes must be declared as public.

44. Where must the package statement appear in a Java source file?

a. At the end of the file
b. Before any class declaration, as the first line
c. After all import statements
d. After the class definition

Ans: b. Before any class declaration, as the first line

Explanation: The package statement must be placed at the very top of the file.

45. What must be true about all classes in a Java package before they can be used?

a. All classes must be compiled before use.
b. Only public classes must be compiled.
c. Only the first class needs to be compiled.
d. Classes in a package do not need to be compiled.

Ans: a. All classes must be compiled before use.

Explanation: Java classes must be compiled into bytecode before they can be used in the program.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

46. Which of the following is true about classes in a Java package?
    a. All classes in the package must have the same name.
    b. All classes in the package must be declared public to be accessed outside the package.
    c. A class in a package cannot be accessed outside unless it is declared static.
    d. A class in a package can be accessed outside the package only if it is declared private.

Ans: b. All classes in the package must be declared public to be accessed outside the package.

Explanation: To access classes outside their package, they must be declared public.

47. What is meant by using the fully qualified name in Java?
    a. Accessing a class using its package name and class name.
    b. Accessing all classes from a package.
    c. Accessing a class using just the class name.
    d. Importing a class without its package name.

Ans: a. Accessing a class using its package name and class name.

Explanation: The fully qualified name is a way to refer to a class by including both its package and class name.

48. Why is the fully qualified name used in Java?
    a. To access all classes from a package at once.
    b. To avoid confusion when two packages have classes with the same name.
    c. To allow access to only the package and not individual classes.
    d. To avoid the need for using the import statement.

Ans: b. To avoid confusion when two packages have classes with the same name.

Explanation: Fully qualified names prevent ambiguity when classes with the same name exist in different packages.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

49. When can you use the fully qualified name in Java?
    a. When accessing a specific class from a package but not the whole package.
    b. Only when the class is public.
    c. Only when the class is static.
    d. When accessing all classes from a package.

Ans: a. When accessing a specific class from a package but not the whole package.

Explanation: Fully qualified names are used to access specific classes when there is no import statement.

50. What is a package inside another package called in Java?
    a. Subclass
    b. Subpackage
    c. Superclass
    d. Namespace

Ans: b.Subpackage

Explanation: A subpackage is a package contained inside another package, creating a hierarchy of packages.

51. Why would you create a subpackage inside a Java package?
    a. To avoid using import statements
    b. To categorize the package further
    c. To combine different packages into one
    d. To store non-class files

Ans: b. To categorize the package further

Explanation: Subpackages allow further organization of related classes and interfaces, making it easier to manage larger projects.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

52. How many import statements can be used in a Java file?

a. Only one
b. Multiple
c. None
d. Only two

Ans: b. Multiple

Explanation: There is no limit to the number of import statements that can be used in a Java file.

53. What is the main purpose of using subpackages in Java?

a. To avoid using classes from the default package
b. To organize and categorize related classes more effectively
c. To prevent naming conflicts
d. To import packages more easily

Ans: b. To organize and categorize related classes more effectively

Explanation: Subpackages provide better organization by grouping related classes and interfaces hierarchically.

54. How many public classes can be there in a Java source file?

a. None
b. One
c. Two
d. Multiple

Ans: b. One

Explanation: A Java source file can contain only one public class, as the file name must match the name of the public class.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

55. What must the name of the Java source file match?
    a. The name of the class with the main method
    b. The name of the public class
    c. The name of the first class in the file
    d. The name of the package

Ans: b. The name of the public class

Explanation: The file name must match the public class name to avoid compilation errors.

56. What happens if there are multiple public classes in a Java source file?
    a. The program will run successfully.
    b. A compilation error will occur.
    c. Only the first class will be compiled.
    d. The program will only compile the public class with the most fields.

Ans: b. A compilation error will occur.

Explanation: Java does not allow more than one public class in a single file, as each public class requires a corresponding file name.

57. If a Java file contains a public class named MyClass, what must the file name be?
    a. MyClass.java
    b. Class.java
    c. publicClass.java
    d. MyClassFile.java

Ans: a. MyClass.java

Explanation: The file name must exactly match the name of the public class for the program to compile.

Name of the Bundle	Advanced Bundle V2	Subject	Java Programming V2
Topic	Package	Last updated on	09 January 2025

58. Why must a Java source file be saved with the name of the public class?

a. To follow the Java naming conventions

b. To avoid compilation errors

c. To allow multiple classes to be compiled

d. To make the code more readable

Ans: b. To avoid compilation errors

Explanation: The Java compiler requires the file name and public class name to match to correctly identify and compile the program.