# Questions

|  |  |
| --- | --- |
| **1.** | package is used for grouping similar kind of classes |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **2.** | package keyword can be anywhere in the file |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **3.** | package keyword should be as a first statement in the java file |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **4.** | how to compile if A.java file which is in pack1 folder with a package as pack1 declaration. | |
| |  | | --- | | A.  javac pack1/A.java |  |  | | --- | | B.  javac pack1.A.java | | | | |
|  | | | |

|  |  |
| --- | --- |
| **5.** | how to run class file(A.class) which is in pack1 package |
| |  | | --- | | A.  java pack1/A |  |  | | --- | | B.  java pack1.A | | | |
|  | | |
| **6.** | package is used for avoiding naming conflicts to the classes |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **7.** | private member of any class should be used in the same class. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **8.** | we cant use private member of one class in another class. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **9.** | we can use private method of A class inside a B class | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | | |
|  | | | | |
| **10.** | | we can use private method of A class inside a B class, if both the classes are in the same java file. | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | | |
|  | | | | |

|  |  |
| --- | --- |
| **11.** | constructor can be private. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **12.** | if class containing private constructor, then object should be created within the same class. | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false |  |  | | --- | | C.  we cant make constructor as a private. | | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **13.** | if class containing only private constror/constructors, then we cant develop a subclass. | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false |  |  | | --- | | C.  we cant make constructor as a private. | | | | |
|  | | | |
| **14.** | private member of a class can be accessed from its subclass. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

|  |  |
| --- | --- |
| **15.** | subclass constructor always should call only super class non private constructor. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false |  |  | | --- | | C.  we cant make constructor as a private. | | | |
|  | | |
| **16.** | private static methods of super class can be accessed from its subclass |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **17.** | outer classes can be private |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **18.** | inner classes can be private |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **19.** | outer classes can't be static |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **20.** | inner classes can't be static |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **21.** | static initialization block can be private |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **22.** | instance initialization block can't be private |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **23.** | private members of super class inheriting to subclass |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **24.** | private members are surviving in the inheritance |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **25.** | default scope also considered as a package |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **26.** | default scoped members are allowed to use in the same package. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **27.** | default scoped members are allowed to use in the other packages also. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **28.** | default scoped members are inheriting to the subclasses of same package. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **29.** | default scoped members are inheriting to the subclasses of any package. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **30.** | protected scoped members are allowed to use in the same package. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **31.** | protected scoped members are allowed to use in the other packages also. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **32.** | protected and default both are one and the same in case of usage |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **33.** | protected scoped member can be inherited to every subclass including subclass of another package | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **34.** | protected scoped member can be inherited to subclass of another package |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

|  |  |
| --- | --- |
| **35.** | if protected member is inheriting to subclass of another package, then it should be used in the within that subclass by using subclass reference type. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **36.** | if protected member is inheriting to subclass of another package, then it should be used in the within that subclass by using super class reference type. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **37.** | if protected member is inheriting to subclass of another package, then it can be used in outside a subclass |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **38.** | if protected member is inheriting to subclass of another package, then it should be used in the same subclass |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **39.** | public scoped members are allowed to use only in the same package. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **40.** | public scoped members are allowed to use in the other packages also. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **41.** | public and protected both are one and the same in case of usage |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **42.** | public scoped member can be inherited to every subclass including subclass of another package | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

|  |  |
| --- | --- |
| **43.** | public scoped member can be inherited to subclass of another package |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **44.** | if public member is inheriting to subclass of another package, then it should be used only in the within that subclass by using subclass reference type. | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **45.** | if public member is inheriting to subclass of another package, then it should be used only in the within that subclass by using super class reference type. | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **46.** | if public member is inheriting to subclass of another package, then it can be used in outside a subclass |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **47.** | if public member is inheriting to subclass of another package, then it should be used only in the same subclass | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **48.** | only public classes can be used in another packages. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

|  |  |
| --- | --- |
| **49.** | we can use one class in another class of the same package without import. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **50.** | we can use another package class without import |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **51.** | package is used for grouping similar kind of classes |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **52.** | package keyword can be anywhere in the file |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **53.** | package keyword should be as a first statement in the java file |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **54.** | how to compile if A.java file which is in pack1 folder with a package as pack1 declaration. | |
| |  | | --- | | A.  javac pack1/A.java |  |  | | --- | | B.  javac pack1.A.java | | | | |
|  | | | |

|  |  |
| --- | --- |
| **55.** | how to run class file(A.class) which is in pack1 package |
| |  | | --- | | A.  java pack1/A |  |  | | --- | | B.  java pack1.A | | | |
|  | | |
| **56.** | package is used for avoiding naming conflicts to the classes |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **57.** | private member of any class should be used in the same class. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **58.** | we cant use private member of one class in another class. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **59.** | we can use private method of A class inside a B class |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **60.** | we can use private method of A class inside a B class, if both the classes are in the same java file. | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

|  |  |
| --- | --- |
| **61.** | constructor can be private. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **62.** | if class containing private constructor, then object should be created within the same class. | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false |  |  | | --- | | C.  we cant make constructor as a private. | | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **63.** | if class containing only private constructor/constructors, then we cant develop a subclass. | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false |  |  | | --- | | C.  we cant make constructor as a private. | | | | |
|  | | | |
| **64.** | private member of a class can be accessed from its subclass. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

|  |  |
| --- | --- |
| **65.** | Sub class constructor always should call only super class non private constructor. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false |  |  | | --- | | C.  we cant make constructor as a private. | | | |
|  | | |
| **66.** | private static methods of super class can be accessed from its subclass |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **67.** | outer classes can be private |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **68.** | inner classes can be private |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **69.** | outer classes can't be static |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **70.** | inner classes can't be static |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **71.** | static initialization block can be private |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **72.** | instance initialization block can't be private |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **73.** | private members of super class inheriting to subclass |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **74.** | private members are surviving in the inheritance |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **75.** | default scope also considered as a package |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **76.** | default scoped members are allowed to use in the same package. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **77.** | default scoped members are allowed to use in the other packages also. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **78.** | default scoped members are inheriting to the subclasses of same package. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **79.** | default scoped members are inheriting to the subclasses of any package. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **80.** | protected scoped members are allowed to use in the same package. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **81.** | protected scoped members are allowed to use in the other packages also. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **82.** | protected and default both are one and the same in case of usage |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **83.** | protected scoped member can be inherited to every subclass including subclass of another package | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **84.** | protected scoped member can be inherited to subclass of another package |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

|  |  |
| --- | --- |
| **85.** | if protected member is inheriting to subclass of another package, then it should be used in the within that subclass by using subclass reference type. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **86.** | if protected member is inheriting to subclass of another package, then it should be used in the within that subclass by using super class reference type. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **87.** | if protected member is inheriting to subclass of another package, then it can be used in outside a subclass |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **88.** | if protected member is inheriting to subclass of another package, then it should be used in the same subclass |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **89.** | public scoped members are allowed to use only in the same package. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |
| **90.** | public scoped members are allowed to use in the other packages also. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | |
|  | | |

|  |  |
| --- | --- |
| **91.** | public and protected both are one and the same in case of usage |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **92.** | public scoped member can be inherited to every subclass including subclass of another package | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

|  |  |
| --- | --- |
| **93.** | public scoped member can be inherited to subclass of another package |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **94.** | if public member is inheriting to subclass of another package, then it should be used only in the within that subclass by using subclass reference type. | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **95.** | if public member is inheriting to subclass of another package, then it should be used only in the within that subclass by using super class reference type. | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **96.** | if public member is inheriting to subclass of another package, then it can be used in outside a subclass |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **97.** | if public member is inheriting to subclass of another package, then it should be used only in the same subclass | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **98.** | only public classes can be used in another packages. |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **99.** | we can use one class in another class of the same package without import. | |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |
| **100.** | | we can use another package class without import |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | | | |
|  | | | |

Bottom of Form