**Core Java Advanced Part – 2**

Object class

  toString (Total available questions : 9)

  equals (Total available questions : 22)

  hashCode (Total available questions : 8)

  Garbage Collection and clone (Total available questions : 6)

  reflection api (Total available questions : 2)

  Enums

  basics (Total available questions : 12)

  advanced (Total available questions : 9)

  Annotations

  annotations (Total available questions : 10)

  Static imports

  static imports (Total available questions : 4)

  Inner classes

  why do we required an inner classes (Total available questions : 3)

  non static, staic and local inner classes (Total available questions : 9)

  anaymous inner class

  lambda expressions (Total available questions : 3)

  Threads

  basics (Total available questions : 114)

  synchronization (Total available questions : 50)

  thread life cycle (Total available questions : 14)

  accessories (Total available questions : 8)

|  |  |
| --- | --- |
| **1.** | in which version of JDK enums are introduced?Q27 |
| |  | | --- | | A.  JDK 1.0 |  |  | | --- | | B.  JDK 1.4 |  |  | | --- | | C.  JDK 1.5 |  |  | | --- | | D.  JDK 1.6 |  |  | | --- | | E.  JDK 1.8 | | |
| **2.** | enums are mainly used for grouping similar kind of and fixed number of constants. true/falseQ28 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |

|  |  |
| --- | --- |
| **3.** | in which scenario enums can't be used?Q29 |
| |  | | --- | | A.  to represent day names |  |  | | --- | | B.  to represent month names |  |  | | --- | | C.  to represent continents |  |  | | --- | | D.  to represent skills | | |
| **4.** | enum can't become a member of a jave fileQ30 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |

|  |  |
| --- | --- |
| **5.** | for enum also class file is generating while compilingQ31 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |
| **6.** | enum constants can be duplicateQ32 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |

|  |  |
| --- | --- |
| **7.** | all enum constants are static by defaultQ33 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |
| **8.** | we can develop an enum inside a class as a member of the classQ34 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |

|  |  |  |
| --- | --- | --- |
| **9.** | which method of enum used for reading all constantsQ35 | |
| |  | | --- | | A.  values() |  |  | | --- | | B.  ordinal() |  |  | | --- | | C.  valueOf() | | | |
| **10.** | | which method of enum used for reading index of a constantQ36 |
| |  | | --- | | A.  values() |  |  | | --- | | B.  ordinal() |  |  | | --- | | C.  valueOf() | | | |

|  |  |
| --- | --- |
| **11.** | which method of enum used for refering a specified constantQ37 |
| |  | | --- | | A.  values() |  |  | | --- | | B.  ordinal() |  |  | | --- | | C.  valueOf() | | |
| **12.** | We can achiev uniformity among the developers with enums.Q38 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **13.** | Every enum should have a minimum one constructor. If we are not developing any constructors, then compiler will provide a default constructorQ39 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **14.** | what is the need of "constant specific class body" (CSCB)?Q40 |
| |  | | --- | | A.  to provide a constructor |  |  | | --- | | B.  to provide a method specific to constant |  |  | | --- | | C.  to provide a method generic to all constant | | |

|  |  |
| --- | --- |
| **15.** | is it possible to choose enum constant as an argument to switch block?Q41 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **16.** | EnumSet is used to group few or all constants of multiple enums.Q42 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **17.** | enum should have a minimum one constant to provide an attribute or a method. or at least there should be semi colon before starting an attribute or a method.Q43 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **18.** | which access level is allowed for constructors of an enumQ44 |
| |  | | --- | | A.  private |  |  | | --- | | B.  protected |  |  | | --- | | C.  public | | |

|  |  |
| --- | --- |
| **19.** | is it possible to develop main method along with the list of constants in the enum?Q45 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **20.** | abstract enum is possible.Q46 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **21.** | java.lang.Enum is super class to all enumsQ47 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **22.** | package com.lara;  public class M1 {  public static void main(String[] args) {  System.out.println(1);  System.out.println(2);  System.out.println(3);  System.out.println(4);  System.out.println(5);  System.out.println(6);  }  }  Q821 |
| |  | | --- | | A.  1  2  3  4  5  6 |  |  | | --- | | B.  1 2 3 4 5 6 | | |

|  |  |
| --- | --- |
| **23.** | by default how many threads are involving in the foreground execution?Q1089 |
| |  | | --- | | A.  1 |  |  | | --- | | A.  2 |  |  | | --- | | C.  3 | | |
| **24.** | In which package Thread class is available?Q1090 |
| |  | | --- | | A.  java.threads |  |  | | --- | | B.  java.lang |  |  | | --- | | C.  java.io | | |

|  |  |
| --- | --- |
| **25.** | Thread is a concrete class.Q1091 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |
| **26.** | We can create an Object to Thread classQ1092 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |

|  |  |
| --- | --- |
| **27.** | Which method of a Thread class we need to override to incorporate a thread task?Q1093 |
| |  | | --- | | A.  start |  |  | | --- | | B.  task |  |  | | --- | | C.  run | | |
| **28.** | Which method of a Thread class we need to call to execute a new thread in the separate thread execution context.Q1094 |
| |  | | --- | | A.  start |  |  | | --- | | B.  task |  |  | | --- | | C.  run | | |

|  |  |
| --- | --- |
| **29.** | Which method of a Thread class we need to call to register a new thread with the thread schedular.Q1095 |
| |  | | --- | | A.  start |  |  | | --- | | B.  task |  |  | | --- | | C.  run | | |
| **30.** | In which package Runnable interface is available?Q1096 |
| |  | | --- | | A.  java.lang |  |  | | --- | | B.  java.io |  |  | | --- | | C.  java.threads | | |

|  |  |
| --- | --- |
| **31.** | How many methods available in the Runnable interfaceQ1097 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  2 |  |  | | --- | | C.  3 |  |  | | --- | | D.  no methods | | |
| **32.** | Runnable interface is a marker interfaceQ1098 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **33.** | Runnable interface is a functional interfaceQ1099 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **34.** | Which method available in the Runnable interface?Q1100 |
| |  | | --- | | A.  start |  |  | | --- | | B.  thread |  |  | | --- | | C.  run | | |

|  |  |
| --- | --- |
| **35.** | Which method we have to implement while developing a subclass to Runnable interface?Q1101 |
| |  | | --- | | A.  start |  |  | | --- | | B.  thread |  |  | | --- | | C.  run | | |
| **36.** | How many ways are there to develop child threads in Java programming language?Q1102 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  2 |  |  | | --- | | C.  3 |  |  | | --- | | D.  no methods | | |

|  |  |
| --- | --- |
| **37.** | In which method we have to implement thread task while developing a child thread by using Runnable interface?Q1103 |
| |  | | --- | | A.  start |  |  | | --- | | B.  thread |  |  | | --- | | C.  run | | |
| **38.** | Which reference type we should suply to the constructor of Thread class while developing a thread by making use of Runnable interface?Q1104 |
| |  | | --- | | A.  ThreadLocal |  |  | | --- | | B.  Number |  |  | | --- | | C.  Runnable | | |

|  |  |
| --- | --- |
| **39.** | start() method is available in the Runnable interface.Q1105 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **40.** | We can call start() method from a class which is subclass to Runnable interface.Q1106 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **41.** | We can call start() method from a class which is subclass to Thread class.Q1107 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **42.** | We can call start() method from a Thread class.Q1108 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **43.** | What happens if main thread is calling run() method from the child thread instead of start() method.Q1109 |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  Runtime Error |  |  | | --- | | C.  Compilation and Running both success and child thread will be allocated to execute run() method., |  |  | | --- | | D.  Compilation and Running both success and main thread will be executing run() method., | | |
| **44.** | if main thread is calling run() method from the child thread instead of start() method, then child will be registring with thread schedular.Q1110 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **45.** | is it possible to call run() method more than once in the same thread reference?Q1111 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **46.** | is it possible to call start() method more than once in the same thread reference?Q1112 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **47.** | what happens while start() method calling more than once in the same thread reference?Q1113 |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  Runtime Error |  |  | | --- | | C.  executing more than one time. | | |
| **48.** | what happens while run() method calling more than once in the same thread reference?Q1114 |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  Runtime Error |  |  | | --- | | C.  executing more than one time. | | |

|  |  |
| --- | --- |
| **49.** | Unhandled Exception object effects only to the current thread.Q1115 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **50.** | Unhandled Exception object effects to all the running threads of current execution.Q1116 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **51.** | If main thread recives unhandled exception, then main thread only terminates from the execution.Q1117 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **52.** | If main thread recives unhandled exception, then all the treads of the current exection terminates.Q1118 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **53.** | If child thread recives unhandled exception, then all the treads of the current exection terminates.Q1119 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **54.** | If child thread recives unhandled exception, then main thread terminates from the execution.Q1120 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **55.** | currentThread() is a static method in the Thread class.Q1121 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **56.** | currentThread() is returning reference of a thread which executes currentThread() method.Q1122 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **57.** | Which t1 refers main thread. (Assume A is a subclass to Thread and B is a subclass to Runnable)Q1123 |
| |  | | --- | | A.  A t1 = new A() |  |  | | --- | | B.  B t1 = new B() |  |  | | --- | | C.  Thread t1 = Thread.currentThread() |  |  | | --- | | D.  Thread t1 = B.currentThread() |  |  | | --- | | E.  Thread t1 = A.currentThread() | | |
| **58.** | Is it possible to define more than one Thread to A class, if A is subclass to Thread class.Q1124 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **59.** | Is it possible to define more than one Thread to B class, if B is subclass to Runnable interfaceQ1125 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **60.** | How many minimum number of objects required to A class to define 2 threads. (Assume A is a subclass to Thread)Q1126 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  2 |  |  | | --- | | C.  we cant define multiple threads to the same class | | |

|  |  |
| --- | --- |
| **61.** | How many minimum number of objects required to A class to define 2 threads. (Assume A is a subclass to Runnable)Q1127 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  2 |  |  | | --- | | C.  we cant define multiple threads to the same class | | |
| **62.** | Every thread is having an unique id.Q1128 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **63.** | what is the data type of id of a threadQ1129 |
| |  | | --- | | A.  int |  |  | | --- | | B.  long |  |  | | --- | | C.  double |  |  | | --- | | D.  String | | |
| **64.** | How to change a thread ID?Q1130 |
| |  | | --- | | A.  set() |  |  | | --- | | B.  setId() |  |  | | --- | | C.  setThreadId() |  |  | | --- | | D.  we cant change | | |

|  |  |
| --- | --- |
| **65.** | How to read a thread ID?Q1131 |
| |  | | --- | | A.  get() |  |  | | --- | | B.  getId() |  |  | | --- | | C.  getThreadId() |  |  | | --- | | D.  we cant read | | |
| **66.** | What is the default name to main thread?Q1132 |
| |  | | --- | | A.  main |  |  | | --- | | B.  initiator |  |  | | --- | | C.  parent |  |  | | --- | | D.  no name | | |

|  |  |
| --- | --- |
| **67.** | What is the default name to 1st child thread?Q1133 |
| |  | | --- | | A.  main |  |  | | --- | | B.  thread-0 |  |  | | --- | | C.  thread-1 |  |  | | --- | | D.  thread-2 | | |
| **68.** | What is the default name to the 2nd child thread?Q1134 |
| |  | | --- | | A.  main |  |  | | --- | | B.  thread-0 |  |  | | --- | | C.  thread-1 |  |  | | --- | | D.  thread-2 | | |

|  |  |
| --- | --- |
| **69.** | Is it possible to change name to main thread?Q1135 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **70.** | how to change default name of a thread?Q1136 |
| |  | | --- | | A.  setName() |  |  | | --- | | B.  setThreadName() |  |  | | --- | | C.  we cant change | | |

|  |  |
| --- | --- |
| **71.** | is it possible to choose a name to child thread while creating?Q1137 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **72.** | what is the miximum priority in case of threads.Q1138 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  5 |  |  | | --- | | C.  10 | | |

|  |  |
| --- | --- |
| **73.** | what is the minimum priority in case of threads.Q1139 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  5 |  |  | | --- | | C.  10 | | |
| **74.** | what is the normal priority in case of threads.Q1140 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  5 |  |  | | --- | | C.  10 | | |

|  |  |
| --- | --- |
| **75.** | What is the value of Thread.MIN\_PRIORITYQ1141 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  5 |  |  | | --- | | C.  10 | | |
| **76.** | What is the value of Thread.NORM\_PRIORITYQ1142 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  5 |  |  | | --- | | C.  10 | | |

|  |  |
| --- | --- |
| **77.** | What is the value of Thread.MAX\_PRIORITYQ1143 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  5 |  |  | | --- | | C.  10 | | |
| **78.** | What is the default priority of a main threadQ1144 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  5 |  |  | | --- | | C.  10 | | |

|  |  |
| --- | --- |
| **79.** | What is the default priority of a child threadQ1145 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  5 |  |  | | --- | | C.  10 |  |  | | --- | | D.  its a priority of its parent thread. | | |
| **80.** | what is the datatype of a priorityQ1146 |
| |  | | --- | | A.  int |  |  | | --- | | B.  long |  |  | | --- | | C.  double |  |  | | --- | | D.  String | | |

|  |  |
| --- | --- |
| **81.** | how to read the priority of a thread Q1147 |
| |  | | --- | | A.  get() |  |  | | --- | | B.  getPriority() |  |  | | --- | | C.  getP() |  |  | | --- | | D.  getInt() | | |
| **82.** | how to change the priority of a thread Q1148 |
| |  | | --- | | A.  set() |  |  | | --- | | B.  setPriority() |  |  | | --- | | C.  setP() |  |  | | --- | | D.  setInt() | | |

|  |  |
| --- | --- |
| **83.** | is it possible to change the priority of a main thread Q1149 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **84.** | if child is depending on parent thread life, then child is called as?Q1150 |
| |  | | --- | | A.  deamon |  |  | | --- | | B.  user | | |

|  |  |
| --- | --- |
| **85.** | if child is not depending on parent thread life, then child is called as?Q1151 |
| |  | | --- | | A.  deamon |  |  | | --- | | B.  user | | |
| **86.** | deamon is a thread which is depending on its parent.Q1152 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **87.** | deamon is a thread which is not depending on its parent.Q1153 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **88.** | user thread is a thread which is depending on its parent.Q1154 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **89.** | user thread is a thread which is not depending on its parent.Q1155 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **90.** | what is the defualt type of a thread in javaQ1156 |
| |  | | --- | | A.  deamon |  |  | | --- | | B.  user | | |

|  |  |
| --- | --- |
| **91.** | which type of a thread creating while developing with a Thread classQ1157 |
| |  | | --- | | A.  deamon |  |  | | --- | | B.  user | | |
| **92.** | which type of a thread creating while developing with a Runnable interfaceQ1158 |
| |  | | --- | | A.  deamon |  |  | | --- | | B.  user | | |

|  |  |
| --- | --- |
| **93.** | how to read a deamon status?Q1159 |
| |  | | --- | | A.  getDeamon() |  |  | | --- | | B.  isDeamon() | | |
| **94.** | how to change a thread from deamon to user (or) from user to deamon?Q1160 |
| |  | | --- | | A.  setDeamon() |  |  | | --- | | B.  changeDeamon() |  |  | | --- | | C.  we cant change | | |

|  |  |
| --- | --- |
| **95.** | When we can change a thread from deamon to user (or) from user to deamon?Q1161 |
| |  | | --- | | A.  before start() |  |  | | --- | | B.  after start() |  |  | | --- | | C.  we cant change | | |
| **96.** | is it possible to change a main thread as a deamon?Q1162 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **97.** | sleep() method in Thread class is a static.Q1163 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **98.** | what is the argument data type of a sleep method in Thread class?Q1164 |
| |  | | --- | | A.  int |  |  | | --- | | B.  long |  |  | | --- | | C.  float |  |  | | --- | | D.  double | | |

|  |  |  |
| --- | --- | --- |
| **99.** | which checked exception required for sleep method in Thread class?Q1165 | |
| |  | | --- | | A.  ArithmaticException |  |  | | --- | | B.  NumberFormatException |  |  | | --- | | C.  InterruptedException |  |  | | --- | | D.  double | | | |
| **100.** | | which checked exception required for join method in Thread class?Q1166 |
| |  | | --- | | A.  ArithmaticException |  |  | | --- | | B.  NumberFormatException |  |  | | --- | | C.  InterruptedException |  |  | | --- | | D.  double | | | |

|  |  |
| --- | --- |
| **101.** | How to tell to parent to wait till child execution got over?Q1167 |
| |  | | --- | | A.  through sleep() |  |  | | --- | | B.  through join() | | |
| **102.** | is it possible to interrupt a thread which is under sleep?Q1168 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **103.** | is it possible to interrupt a thread which is under join?Q1169 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **104.** | whih method is used to interrupt while thread is under sleep?Q1170 |
| |  | | --- | | A.  interrupt() |  |  | | --- | | B.  makeStop() |  |  | | --- | | C.  stopInterrupt() | | |

|  |  |
| --- | --- |
| **105.** | whih method is used to interrupt while thread is under join?Q1171 |
| |  | | --- | | A.  interrupt() |  |  | | --- | | B.  makeStop() |  |  | | --- | | C.  stopInterrupt() | | |
| **106.** | How many maximum locks will be for one object?Q1172 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  2 |  |  | | --- | | C.  it depends on how many super classes are there? | | |

|  |  |
| --- | --- |
| **107.** | How many maximum locks will be for one class?Q1173 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  2 |  |  | | --- | | C.  it depends on how many super classes are there? | | |
| **108.** | Which lock is required while accessing synchronized and non static methodQ1174 |
| |  | | --- | | A.  class lock |  |  | | --- | | B.  object lock |  |  | | --- | | C.  no lock is required | | |

|  |  |
| --- | --- |
| **109.** | Which lock is required while accessing synchronized and static methodQ1175 |
| |  | | --- | | A.  class lock |  |  | | --- | | B.  object lock |  |  | | --- | | C.  no lock is required | | |
| **110.** | Which lock is required while accessing non synchronized and static methodQ1176 |
| |  | | --- | | A.  class lock |  |  | | --- | | B.  object lock |  |  | | --- | | C.  no lock is required | | |

|  |  |
| --- | --- |
| **111.** | Which lock is required while accessing non synchronized and non static methodQ1177 |
| |  | | --- | | A.  class lock |  |  | | --- | | B.  object lock |  |  | | --- | | C.  no lock is required | | |
| **112.** | to access a synchronized and non static method on a1, thread should required \_\_\_\_Q1178 |
| |  | | --- | | A.  object lock of any object |  |  | | --- | | B.  object lock of an object which is refered by a1 |  |  | | --- | | C.  object lock of an object which is refered by thread reference | | |

|  |  |
| --- | --- |
| **113.** | Assume a1 is a type of A class and it is pointing to an object of A class. To access a synchronized and non static method on a1, thread should required \_\_\_\_Q1179 |
| |  | | --- | | A.  object lock of any object |  |  | | --- | | B.  object lock of an object which is refered by a1 |  |  | | --- | | C.  object lock of an object which is refered by thread reference |  |  | | --- | | D.  class lock of A class. | | |
| **114.** | Assume a1 is a type of A class and it is pointing to an object of A class. To access a synchronized and static method on a1, thread should required \_\_\_\_Q1180 |
| |  | | --- | | A.  object lock of any object |  |  | | --- | | B.  object lock of an object which is refered by a1 |  |  | | --- | | C.  object lock of an object which is refered by thread reference |  |  | | --- | | D.  class lock of A class. | | |

|  |  |
| --- | --- |
| **115.** | Assume a1 is a type of A class and it is pointing to an object of A class. Is it possible to access two threads any nonstatic and non synchronized method on a1 simultaniously?Q1181 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **116.** | Assume a1 is a type of A class and it is pointing to an object of A class. Is it possible to access two threads any nonstatic and non synchronized different methods on a1 simultaniously?Q1182 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **117.** | Assume a1 is a type of A class and it is pointing to an object of A class. Is it possible to access two threads any synchronized and nonstatic method on a1 simultaniously?Q1183 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **118.** | Assume a1 is a type of A class and it is pointing to an object of A class. Is it possible to access two threads any synchronized and nonstatic different methods on a1 simultaniously?Q1184 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **119.** | Assume a1 is a type of A class and it is pointing to an object of A class and a2 is a type of A class and it is pointing to another object of same A class . Assume A class containing synchronized and non static test1() method. Is it possible, one thread accessing a1.test1() and another thread accessing a2.test1() simultaniously?Q1185 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **120.** | Assume a1 is a type of A class and it is pointing to an object of A class and a2 is a type of A class and it is pointing to another object of same A class . Assume A class containing synchronized and non static test1() and test2() method2. Is it possible, one thread accessing a1.test1() and another thread accessing a2.test2() simultaniously?Q1186 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **121.** | Assume a1 is a type of A class and it is pointing to an object of A class. Assume A class containing synchronized and non static test1() and test2() method2. Is it possible, one thread accessing a1.test1() and another thread accessing a1.test2() simultaniously?Q1187 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **122.** | Assume A class containing static and synchronized test1() method. Is it possible, two threads accessing A.test1() simultaniously?Q1188 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **123.** | Assume A class containing static and synchronized test1() and test2() methods. Is it possible, one thread accessing A.test1() and another thread accessing A.test2() simultaniously?Q1189 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **124.** | while thread is entering into synchronized and non static method on a1 pointing object, then it should required object lock of an object which is refered by a1.Q1190 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **125.** | If thread required object lock of current object to enter into synchronized block, then what could be the mutext to the synchronized block?Q1191 |
| |  | | --- | | A.  Runnable reference |  |  | | --- | | B.  Thread reference |  |  | | --- | | C.  this | | |
| **126.** | if t1 thread holding object lock of a1 and looking for a2 object lock. and if t2 thread holding object lock a2 and looking for a1 object lock. Then which thread will execute first.Q1192 |
| |  | | --- | | A.  t1 |  |  | | --- | | B.  t2 |  |  | | --- | | C.  dead lock | | |

|  |  |
| --- | --- |
| **127.** | In which class wait() method is developed?Q1193 |
| |  | | --- | | A.  Thread |  |  | | --- | | B.  Object |  |  | | --- | | C.  ThreadLocal | | |
| **128.** | In which class notify() method is developed?Q1194 |
| |  | | --- | | A.  Thread |  |  | | --- | | B.  Object |  |  | | --- | | C.  ThreadLocal | | |

|  |  |
| --- | --- |
| **129.** | In which class notifyAll() method is developed?Q1195 |
| |  | | --- | | A.  Thread |  |  | | --- | | B.  Object |  |  | | --- | | C.  ThreadLocal | | |
| **130.** | String class having wait() method.Q1196 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **131.** | Thread class having notify() method.Q1197 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **132.** | Thread should required object lock of t1 to call t1.wait()Q1198 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **133.** | Thread should required object lock of t1 to call t1.notify()Q1199 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **134.** | Thread should required object lock of t2 to call t1.notifyAll()Q1200 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **135.** | Thread having object lock of t1 and trying to call t2.wait(). t1 and t2 are refering to different objects. Now thread will go to waiting state or not?Q1201 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **136.** | which checked exception required to call wait() method?Q1202 |
| |  | | --- | | A.  ClassNotFoundException |  |  | | --- | | B.  SQLException |  |  | | --- | | C.  InterruptedException | | |

|  |  |
| --- | --- |
| **137.** | main thread is about to call a1.wait() in the main method by choosing synchronized block. What could be the mutext for the synchronized block?Q1203 |
| |  | | --- | | A.  this |  |  | | --- | | B.  main thread reference |  |  | | --- | | C.  a1 | | |
| **138.** | child thread is about to call wait() in the run method. by choosing synchronized block. What could be the mutext for the synchronized block?Q1204 |
| |  | | --- | | A.  this |  |  | | --- | | B.  main thread reference |  |  | | --- | | C.  a1 | | |

|  |  |
| --- | --- |
| **139.** | main thread is about to call a1.notify() in the main method by choosing synchronized block. What could be the mutext for the synchronized block?Q1205 |
| |  | | --- | | A.  this |  |  | | --- | | B.  main thread reference |  |  | | --- | | C.  a1 | | |
| **140.** | child thread is about to call notifyAll() in the run method. by choosing synchronized block. What could be the mutext for the synchronized block?Q1206 |
| |  | | --- | | A.  this |  |  | | --- | | B.  main thread reference |  |  | | --- | | C.  a1 | | |

|  |  |
| --- | --- |
| **141.** | there are two objects. t1 is refering to one object and t2 is refering to another object. child1 went into waiting by calling wait on t1. how child1 can be released from the waiting?Q1207 |
| |  | | --- | | A.  some one has to call t1.notify() |  |  | | --- | | B.  some one has to call t2.notifyAll() | | |
| **142.** | there are two objects. t1 is refering to one object and t2 is refering to another object. some child threads went into waiting by calling wait on t1. how all child threads can be released from the waiting?Q1208 |
| |  | | --- | | A.  some one has to call t1.notify() |  |  | | --- | | B.  some one has to call t2.notifyAll() |  |  | | --- | | C.  some one has to call t1.notifyAll() | | |

|  |  |
| --- | --- |
| **143.** | while thread is going to wait stage, it relases owned lock.Q1209 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |
| **144.** | while thread is calling join, it relases owned lock.Q1210 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |

|  |  |
| --- | --- |
| **145.** | while thread is calling sleep, it relases owned lock.Q1211 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |
| **146.** | ThreadLocal maintains a value local to a perticular thread.Q1212 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |

|  |  |
| --- | --- |
| **147.** | if same ThreadLocal object is using by two different threads and first thread set a value as 10 and 2nd thread not set a value yet. Then what could be the value while reading by 2nd thread.Q1213 |
| |  | | --- | | A.  10 |  |  | | --- | | B.  0 |  |  | | --- | | C.  null |  |  | | --- | | D.  exception | | |
| **148.** | is it possibe to stop all the threads which are under one group by using group reference?Q1214 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **149.** | ThreadGroup is used for grouping any number of and any type of threads.Q1215 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |
| **150.** | what is the state of a thread after calling sleep method.Q1216 |
| |  | | --- | | A.  WAITING |  |  | | --- | | B.  TIMED\_WAITING |  |  | | --- | | C.  RUNNABLE |  |  | | --- | | D.  NEW |  |  | | --- | | E.  BLOCKED | | |

|  |  |
| --- | --- |
| **151.** | what is the state of a thread after calling wait method.Q1217 |
| |  | | --- | | A.  WAITING |  |  | | --- | | B.  TIMED\_WAITING |  |  | | --- | | C.  RUNNABLE |  |  | | --- | | D.  NEW |  |  | | --- | | E.  BLOCKED | | |
| **152.** | what is the state of a thread after calling join method.Q1218 |
| |  | | --- | | A.  WAITING |  |  | | --- | | B.  TIMED\_WAITING |  |  | | --- | | C.  RUNNABLE |  |  | | --- | | D.  NEW |  |  | | --- | | E.  BLOCKED | | |

|  |  |
| --- | --- |
| **153.** | what is the state of a thread after calling start method while executing run method..Q1219 |
| |  | | --- | | A.  WAITING |  |  | | --- | | B.  TIMED\_WAITING |  |  | | --- | | C.  RUNNABLE |  |  | | --- | | D.  NEW |  |  | | --- | | E.  BLOCKED | | |
| **154.** | what is the state of a thread before calling start methodQ1220 |
| |  | | --- | | A.  WAITING |  |  | | --- | | B.  TIMED\_WAITING |  |  | | --- | | C.  RUNNABLE |  |  | | --- | | D.  NEW |  |  | | --- | | E.  BLOCKED | | |

|  |  |
| --- | --- |
| **155.** | what is the state of a thread while it is in dead lockQ1221 |
| |  | | --- | | A.  WAITING |  |  | | --- | | B.  TIMED\_WAITING |  |  | | --- | | C.  RUNNABLE |  |  | | --- | | D.  NEW |  |  | | --- | | E.  BLOCKED | | |
| **156.** | what is the state of a thread once its execution got overQ1222 |
| |  | | --- | | A.  WAITING |  |  | | --- | | B.  TIMED\_WAITING |  |  | | --- | | C.  RUNNABLE |  |  | | --- | | D.  TERMINATED |  |  | | --- | | E.  BLOCKED | | |

|  |  |
| --- | --- |
| **157.** | yield() is used for requesting current thread to give a chance to another threadsQ1223 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **158.** | In which version of JDK, annotations are introduced?Q1423 |
| |  | | --- | | A.  JDK1.1 |  |  | | --- | | B.  JDK1.3 |  |  | | --- | | C.  JDK1.4 |  |  | | --- | | D.  JDK1.5 |  |  | | --- | | E.  JDK1.8 | | |

|  |  |
| --- | --- |
| **159.** | annotation method cant have a return type ofQ1424 |
| |  | | --- | | A.  int |  |  | | --- | | B.  void |  |  | | --- | | C.  String |  |  | | --- | | D.  Integer |  |  | | --- | | E.  boolean[] | | |
| **160.** | annotations are used for providing a description. Q1425 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |

|  |  |
| --- | --- |
| **161.** | while using an annotation, we can skip the values to any method.Q1426 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |
| **162.** | while using an annotation, we can skip the values to any method.Q1427 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |

|  |  |
| --- | --- |
| **163.** | annotation method can have a default value as a return value.Q1428 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |
| **164.** | which meta annotation is used to specify usage limited to perticular type of members?Q1429 |
| |  | | --- | | A.  annotation |  |  | | --- | | B.  Retentin |  |  | | --- | | C.  Target | | |

|  |  |
| --- | --- |
| **165.** | which is not a RetentionPolicyQ1430 |
| |  | | --- | | A.  JAVA |  |  | | --- | | B.  CLASS |  |  | | --- | | C.  RUNTIME | | |
| **166.** | while calling add method of ArrayList which type of warning can be expected?Q1431 |
| |  | | --- | | A.  deprecation |  |  | | --- | | B.  unchecked |  |  | | --- | | C.  supress | | |

|  |  |
| --- | --- |
| **167.** | how to supress dectation and unchecked messages both?Q1432 |
| |  | | --- | | A.  @SupressWarnings({"deprecation", "unchecked"}) |  |  | | --- | | B.  @SupressWarnings("deprecation", "unchecked") | | |
| **168.** | package toString;  class A{  int i;  }  public class M1 {  public static void main(String[] args) {  A a1 = new A();  a1.i = 20;  System.out.println(a1);  }  }  Q1847 |
| |  | | --- | | A.  toString.A@15db9742 |  |  | | --- | | B.  20 |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **169.** | package toString.pack1;  class A{  int i;  public String toString() {  return "its A type object with i value as: " + i;  }  }  public class M1 {  public static void main(String[] args) {  A a1 = new A();  a1.i = 20;  System.out.println(a1);  }  }  Q1848 |
| |  | | --- | | A.  its A type object with i value as: 20 |  |  | | --- | | B.  20 |  |  | | --- | | C.  toString.pack1.A@15db9742 |  |  | | --- | | D.  None | | |
| **170.** | package toString.pack1;  class A{  int i;  }  public class M1 {  public static void main(String[] args) {  A a1 = new A();  a1.i = 20;  System.out.println(a1);  String s1 = "desc: " + a1;  System.out.println(s1);  }  }Q1849 |
| |  | | --- | | A.  its A type object with i value as: 20  desc: |  |  | | --- | | B.  its A type object with i value as: 20  desc: its A type object with i value as: 20 |  |  | | --- | | C.  toString.pack1.A@15db9742  desc: toString.pack1.A@15db9742 |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **171.** | package toString.pack1;  class A{  int i;  public String toString() {  return "its A type object with i value as: " + i;  }  }  public class M1 {  public static void main(String[] args) {  A a1 = new A();  a1.i = 20;  System.out.println(a1);  String s1 = "desc: " + a1;  System.out.println(s1);  }  }  Q1850 |
| |  | | --- | | A.  its A type object with i value as: 20  desc: |  |  | | --- | | B.  its A type object with i value as: 20  desc: its A type object with i value as: 20 |  |  | | --- | | C.  toString.pack1.A@15db9742  desc: toString.pack1.A@15db9742 |  |  | | --- | | D.  None | | |
| **172.** | package toString.pack1;  class B{  int i, j;  B(int i, int j){  this.i = i;  this.j = j;  }  }  public class M2 {  public static void main(String[] args) {  B b1 = new B(10, 20);  System.out.println(b1);  }  }  Q1851 |
| |  | | --- | | A.  toString.pack1.B@15db9742 |  |  | | --- | | B.  10, 20 |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **173.** | package toString.pack1;  class B{  int i, j;  B(int i, int j){  this.i = i;  this.j = j;  }  public String toString() {  return "Its from B class i: " + i + ", j: " + j;  }  }  public class M2 {  public static void main(String[] args) {  B b1 = new B(10, 20);  System.out.println(b1);  }  }  Q1852 |
| |  | | --- | | A.  toString.pack1.B@15db9742 |  |  | | --- | | B.  10, 20 |  |  | | --- | | C.  Its from B class i: 10, j: 20 |  |  | | --- | | D.  None | | |
| **174.** | package toString.pack1;  class C{  int i;  }  class D{  int j;  C c1;  }  public class M3 {  public static void main(String[] args) {  C c1 = new C();  c1.i = 10;  D d1 = new D();  d1.j = 20;  d1.c1 = c1;  System.out.println(c1);  System.out.println(d1);  }  }  Q1853 |
| |  | | --- | | B.  toString.pack1.C@15db9742  toString.pack1.D@6d06d69c |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **175.** | package toString.pack1;  class C{  int i;  @Override  public String toString() {  // TODO Auto-generated method stub  return "i = " + i;  }  }  class D{  int j;  C c1;  @Override  public String toString() {  // TODO Auto-generated method stub  return "j = " + j + ", " + c1;  }  }  public class M3 {  public static void main(String[] args) {  C c1 = new C();  c1.i = 10;  D d1 = new D();  d1.j = 20;  d1.c1 = c1;  System.out.println(c1);  System.out.println(d1);  }  }  Q1854 |
| |  | | --- | | A.  i = 10  j = 20, i = 10 |  |  | | --- | | B.  toString.pack1.C@15db9742  j = 20, toString.pack1.C@15db9742 |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |
| **176.** | package toString.pack1;  import java.util.ArrayList;  public class M5 {  public static void main(String[] args) {  String s1 = "Hello";  Integer obj = 10;  Thread t1 = new Thread();  ArrayList list = new ArrayList();  System.out.println(s1);  System.out.println(obj);  System.out.println(t1);  System.out.println(list);  }  }  Q1855 |
| |  | | --- | | A.  Hello  10  Thread[Thread-0,5,main]  [] |  |  | | --- | | B.  Compilation Error |  |  | | --- | | C.  hello  10  Exception | | |

|  |  |
| --- | --- |
| **177.** | package toString.pack1;  class G{  int i;  }  public class M6 {  public static void main(String[] args) {  G g1 = new G();  g1.i = 10;    G g2 = new G();  g2.i = 10;  System.out.println(g1 == g2);  System.out.println(g1.equals(g2));  }  }  Q1856 |
| |  | | --- | | A.  true  true |  |  | | --- | | B.  false  false |  |  | | --- | | C.  true  false |  |  | | --- | | D.  false  true | | |
| **178.** | package toString.pack1;  class G{  int i;  public boolean equals(Object obj) {  return i == ((G) obj).i;  }  }  public class M6 {  public static void main(String[] args) {  G g1 = new G();  g1.i = 10;    G g2 = new G();  g2.i = 10;  System.out.println(g1 == g2);  System.out.println(g1.equals(g2));    G g3 = g1;  System.out.println(g3 == g1);  System.out.println(g3.equals(g1));  }  }  Q1857 |
| |  | | --- | | A.  false  true  true  true |  |  | | --- | | B.  true  false  true  false |  |  | | --- | | C.  true  true  true  true |  |  | | --- | | D.  Compilation Error | | |

|  |  |
| --- | --- |
| **179.** | package toString.pack2;  class A{  int i;  }  public class M1 {  public static void main(String[] args) {  A a1 = new A();  a1.i = 10;    A a2 = new A();  a2.i = 10;    System.out.println(a1 == a2);  System.out.println(a1.equals(a2));  System.out.println(a1.i == a2.i);  }  }  Q1858 |
| |  | | --- | | A.  false  false  true |  |  | | --- | | B.  true  true  false |  |  | | --- | | C.  true  true  true | | |
| **180.** | package toString.pack2;  class A{  int i;  }  public class M1 {  public static void main(String[] args) {  A a1 = new A();  a1.i = 10;    A a2 = new A();  a2.i = 10;    System.out.println(a1 == a2);  System.out.println(a1.equals(a2));  System.out.println(a1.i == a2.i);    A a3 = a1;  System.out.println(a1 == a3);  System.out.println(a1.equals(a3));  System.out.println(a1.i == a3.i);  }  }  Q1859 |
| |  | | --- | | A.  true  true  true  false  false  true |  |  | | --- | | B.  false  false  true  true  true  true |  |  | | --- | | C.  true  true  true  true  true  true |  |  | | --- | | D.  false  false  false  false  false  false | | |

|  |  |
| --- | --- |
| **181.** | package toString.pack2;  class B{  int i;  double j;  B(int i, double j) {  this.i = i;  this.j = j;  }  }  public class M2 {  public static void main(String[] args) {  B b1 = new B(10, 2.5);  B b2 = new B(10, 2.5);      System.out.println(b1 == b2);  System.out.println(b1.equals(b2));  System.out.println(b1.i == b2.i && b1.j == b2.j);  }  }  Q1860 |
| |  | | --- | | A.  true  false  true |  |  | | --- | | B.  true  true  false |  |  | | --- | | C.  false  false  true | | |
| **182.** | package toString.pack2;  class B{  int i;  double j;  B(int i, double j) {  this.i = i;  this.j = j;  }  }  public class M2 {  public static void main(String[] args) {  B b1 = new B(10, 2.5);  B b2 = new B(10, 2.5);  B b3 = b2;    System.out.println(b1 == b2);  System.out.println(b1.equals(b2));  System.out.println(b1.i == b2.i && b1.j == b2.j);    System.out.println(b3 == b2);  System.out.println(b3.equals(b2));  System.out.println(b3.i == b2.i && b3.j == b2.j);  }  }  Q1861 |
| |  | | --- | | A.  true  true  true  false  false  true |  |  | | --- | | B.  false  false  true  true  true  true |  |  | | --- | | C.  true  true  true  true  true  true |  |  | | --- | | D.  false  false  false  false  false  false | | |

|  |  |
| --- | --- |
| **183.** | package toString.pack2;  class C  {  int i;  C(int i) {  this.i = i;  }  public boolean equals(Object obj) {  return (this.i == ((C) obj).i);  }  }  public class M3 {  public static void main(String[] args) {  C c1 = new C(90);  C c2 = new C(90);  System.out.println(c1 == c2);  System.out.println(c1.equals(c2));  System.out.println(c1.i == c2.i);  }  }  Q1862 |
| |  | | --- | | A.  true  false  true |  |  | | --- | | B.  true  true  false |  |  | | --- | | C.  false  false  true |  |  | | --- | | D.  false  true  true | | |
| **184.** | package toString.pack2;  class D{  int i, j;  D(int i, int j) {  this.i = i;  this.j = j;  }  @Override  public boolean equals(Object obj) {  // TODO Auto-generated method stub  return (this.i == ((D)obj).i && this.j == ((D)obj).j);  }  }  public class M4 {  public static void main(String[] args) {  D d1 = new D(10,20);  D d2 = new D(10,20);    System.out.println(d1 == d2);  System.out.println(d1.equals(d2));  System.out.println(d1.i == d2.j && d1.j == d2.j);  }  }  Q1863 |
| |  | | --- | | A.  true  false  true |  |  | | --- | | B.  true  true  false |  |  | | --- | | C.  false  false  true |  |  | | --- | | D.  false  true  true | | |

|  |  |
| --- | --- |
| **185.** | package toString.pack2;  class E  {  int i, j;  double k;  E(int i, int j, double k){  this.i = i;  this.j = j;  this.k = k;  }  @Override  public boolean equals(Object obj) {  E ref = (E) obj;  boolean flag = (i == ref.i &&  j == ref.j &&  k == ref.k);  return flag;  }  }  public class M5 {  public static void main(String[] args) {  E e1 = new E(10, 20, 5.5);  E e2 = new E(10, 20, 5.5);  E e3 = new E(10, 20, 5.4);    System.out.println(e1.equals(e2));  System.out.println(e1.equals(e3));  System.out.println(e2.equals(e3));  }  }  Q1864 |
| |  | | --- | | A.  true  false  false |  |  | | --- | | B.  true  true  false |  |  | | --- | | C.  false  false  true |  |  | | --- | | D.  false  true  true | | |
| **186.** | package toString.pack2;  class F  {  int i;  F(int i){  this.i = i;  }  @Override  public boolean equals(Object obj) {  F ref = (F) obj;  return i == ref.i;  }  }  public class M6 {  public static void main(String[] args) {  F f1 = new F(10);  F f2 = new F(10);    A a1 = new A();  a1.i = 10;    System.out.println(f1.equals(f2));  System.out.println(f1.equals(a1));  }  }  Q1865 |
| |  | | --- | | A.  true  ClassCastException |  |  | | --- | | B.  false  ClassCastException |  |  | | --- | | C.  true  false | | |

|  |  |
| --- | --- |
| **187.** | package toString.pack2;  class F  {  int i;  F(int i){  this.i = i;  }  @Override  public boolean equals(Object obj) {  if( ! (obj instanceof F)) {  return false;  }  F ref = (F) obj;  return i == ref.i;  }  }  public class M6 {  public static void main(String[] args) {  F f1 = new F(10);  F f2 = new F(10);    A a1 = new A();  a1.i = 10;    System.out.println(f1.equals(f2));  System.out.println(f1.equals(a1));  }  }  Q1866 |
| |  | | --- | | A.  true  true |  |  | | --- | | B.  true  false |  |  | | --- | | C.  false  true |  |  | | --- | | D.  false  false | | |
| **188.** | package toString.pack2;  class G{  int i;  G(int i){  this.i = i;  }  @Override  public boolean equals(Object obj) {  return (obj instanceof G && i == ((G)obj).i);  }  }  public class M7 {  public static void main(String[] args) {  G g1 = new G(10);  F f1 = new F(10);  A a1 = new A();  a1.i = 10;    System.out.println(g1.equals(f1));  System.out.println(g1.equals(a1));  System.out.println(g1.equals(10));  System.out.println(g1.equals(10.0));  }  }  Q1867 |
| |  | | --- | | A.  false  true  true  true |  |  | | --- | | B.  true  false  true  false |  |  | | --- | | C.  true  true  true  true |  |  | | --- | | D.  false  false  false  false | | |

|  |  |
| --- | --- |
| **189.** | package toString.pack2;  class H  {  int i, j, k;  H(int i, int j, int k){  this.i = i;  this.j = j;  this.k = k;  }  @Override  public boolean equals(Object obj) {  return (obj instanceof H &&  i == ((H) obj).i &&  j == ((H)obj).j &&  k == ((H)obj).k);  }  }  public class M8 {  public static void main(String[] args) {  H h1 = new H(10, 20, 30);  H h2 = new H(10, 20, 30);  System.out.println(h1.equals(h2));  System.out.println(h1.equals(10));  System.out.println(h1.equals(new H(10, 20, 30)));  System.out.println(h1.equals(new H(10, 20, 31)));  System.out.println(h1.equals("hello"));  }  }  Q1868 |
| |  | | --- | | A.  true  false  true  false  false |  |  | | --- | | B.  true  false  true  false  true |  |  | | --- | | C.  true  true  true  false  false | | |
| **190.** | package toString.pack2;  public class M9 {  public static void main(String[] args) {  String s1 = "hello";  String s2 = "hello";  String s3 = new String("hello");    System.out.println(s1.equals(s2));  System.out.println(s1.equals(s3));  System.out.println(s2.equals(s3));  }  }  Q1869 |
| |  | | --- | | A.  true  false  false |  |  | | --- | | B.  true  true  true |  |  | | --- | | C.  false  false  true |  |  | | --- | | D.  false  true  true | | |

|  |  |
| --- | --- |
| **191.** | package toString.pack2;  public class M10 {  public static void main(String[] args) {  Integer obj1 = new Integer(90);  Integer obj2 = new Integer(90);  Integer obj3 = 90;  System.out.println(obj1.equals(obj2));  System.out.println(obj1.equals(obj3));  System.out.println(obj2.equals(obj3));  }  }  Q1870 |
| |  | | --- | | A.  true  false  false |  |  | | --- | | B.  true  true  true |  |  | | --- | | C.  false  false  true |  |  | | --- | | D.  false  true  true | | |
| **192.** | package toString.pack2;  public class M11 {  public static void main(String[] args) {  StringBuffer sb1 = new StringBuffer("hello");  StringBuffer sb2 = new StringBuffer("hello");  System.out.println(sb1.equals(sb2));  }  }  Q1871 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |

|  |  |
| --- | --- |
| **193.** | package toString.pack2;  public class M12 {  public static void main(String[] args) {  StringBuilder sb1 = new StringBuilder("hello");  StringBuilder sb2 = new StringBuilder("hello");  System.out.println(sb1.equals(sb2));  }  }  Q1872 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |
| **194.** | package toString.pack2;  import java.util.ArrayList;  public class M13 {  public static void main(String[] args) {  ArrayList<Integer> list1 = new ArrayList<Integer>();  list1.add(90);  list1.add(910);  list1.add(190);    ArrayList<Integer> list2 = new ArrayList<Integer>();  list2.add(90);  list2.add(910);  list2.add(190);    System.out.println(list1.equals(list2));  }  }  Q1873 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |

|  |  |
| --- | --- |
| **195.** | package toString.pack2;  class I  {  int x;  String s1;  @Override  public boolean equals(Object obj) {  return (obj instanceof I && (x == ((I)obj).x) && (s1.equals(((I)obj).s1)));  }  }  public class M14 {  public static void main(String[] args) {  I obj1 = new I();  obj1.x = 10;  obj1.s1 = "hello";    I obj2 = new I();  obj2.x = 10;  obj2.s1 = "hello";    System.out.println(obj1.equals(obj2));  }  }  Q1874 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |
| **196.** | package toString.pack2;  class I  {  int x;  String s1;  @Override  public boolean equals(Object obj) {  return (obj instanceof I && (x == ((I)obj).x) && (s1.equals(((I)obj).s1)));  }  }  public class M14 {  public static void main(String[] args) {  I obj1 = new I();  obj1.x = 10;  obj1.s1 = "hello";    I obj2 = new I();  obj2.x = 10;  obj2.s1 = "hello";    System.out.println(obj1.equals(obj2));    I obj3 = new I();  System.out.println(obj1.equals(obj3));  }  }  Q1875 |
| |  | | --- | | A.  true  true |  |  | | --- | | B.  true  false |  |  | | --- | | C.  false  false | | |

|  |  |
| --- | --- |
| **197.** | package toString.pack2;  class I  {  int x;  String s1;  @Override  public boolean equals(Object obj) {  return (obj instanceof I && (x == ((I)obj).x) && (s1.equals(((I)obj).s1)));  }  }  public class M14 {  public static void main(String[] args) {  I obj1 = new I();  obj1.x = 10;  obj1.s1 = "hello";    I obj2 = new I();  obj2.x = 10;  obj2.s1 = "hello";    System.out.println(obj1.equals(obj2));    I obj3 = new I();  obj3.x = 10;  System.out.println(obj3.equals(obj1));  }  }  Q1876 |
| |  | | --- | | A.  true  true |  |  | | --- | | B.  true  false |  |  | | --- | | C.  false  false |  |  | | --- | | D.  NullPointerException | | |
| **198.** | package toString.pack2;  class J  {  int x;  String s1;  public boolean equals(Object obj) {  boolean flag = (obj instanceof J) &&  (x == ((J)obj).x) &&  (  s1 == null?  (((J)obj).s1 == null ? true : false) :  (((J)obj).s1 == null ? false : s1.equals(((J)obj).s1))  );  return flag;  }  }  public class M15 {  public static void main(String[] args) {  J j1 = new J();  j1.x = 10;  J j2 = new J();  J j3 = new J();  j3.x = 10;  System.out.println(j1.equals(j2));  System.out.println(j1.equals(j3));  System.out.println(j2.equals(j3));  System.out.println(j3.equals(j1));  System.out.println(j3.equals(j2));  System.out.println(j2.equals(j1));  }  }  Q1877 |
| |  | | --- | | A.  false  true  false  true  false  false |  |  | | --- | | B.  false  true  false  true  false  true |  |  | | --- | | C.  false  false  false  true  false  false | | |

|  |  |
| --- | --- |
| **199.** | package toString.pack3;  class A  {  int i;  A(int i){  this.i = i;  }  }  public class M1 {  public static void main(String[] args) {  A a1 = new A(90);  A a2 = new A(90);  System.out.println(a1.hashCode());  System.out.println(a2.hashCode());  A a3 = a1;  System.out.println(a3.hashCode());  }  }  //Check which two objects hashcode values are equalQ1878 |
| |  | | --- | | A.  a1 and a3 |  |  | | --- | | B.  a1 and a2 |  |  | | --- | | C.  a2 and a3 | | |
| **200.** | package toString.pack3;  class B  {  int i;  int j;  B(int i, int j){  this.i = i;  this.j = j;  }  }  public class M2 {  public static void main(String[] args) {  B b1 = new B(90, 10);  B b2 = new B(90, 10);  B b3 = b2;  System.out.println(b1.hashCode());  System.out.println(b2.hashCode());  System.out.println(b3.hashCode());  }  }  //Check which two objects hashcode values are equalQ1879 |
| |  | | --- | | A.  a1 and a3 |  |  | | --- | | B.  a1 and a2 |  |  | | --- | | C.  a2 and a3 | | |

|  |  |
| --- | --- |
| **201.** | package toString.pack3;  class C  {  int i;  @Override  public int hashCode() {  return i;  }  }  public class M3 {  public static void main(String[] args) {  C c1 = new C();  c1.i = 10;  C c2 = new C();  c2.i = 10;    System.out.println(c1.hashCode());  System.out.println(c2.hashCode());  }  }  Q1880 |
| |  | | --- | | A.  10  10 |  |  | | --- | | B.  366712642  1829164700 |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |
| **202.** | package toString.pack3;  public class M4 {  public static void main(String[] args) {  String s1 = "hello";  String s2 = "hello";  String s3 = new String("hello");  System.out.println(s1.hashCode());  System.out.println(s2.hashCode());  System.out.println(s3.hashCode());  }  }  // Check whether hashcode values are same or differentQ1881 |
| |  | | --- | | A.  Different |  |  | | --- | | B.  Same | | |

|  |  |
| --- | --- |
| **203.** | package toString.pack3;  public class M7 {  public static void main(String[] args) {  Integer obj1 = 100;  Integer obj2 = new Integer(100);  System.out.println(obj1.hashCode());  System.out.println(obj2.hashCode());  }  }  // what is the output hereQ1882 |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  100  100 |  |  | | --- | | C.  hashcode values | | |
| **204.** | package toString.pack3;  import java.util.ArrayList;  public class M8 {  public static void main(String[] args) {  ArrayList<String> list1 = new ArrayList<String>();  list1.add("hello1");  list1.add("hello2");  list1.add("hello3");    ArrayList<String> list2 = new ArrayList<String>();  list2.add("hello1");  list2.add("hello2");  list2.add("hello3");    System.out.println(list1.hashCode());  System.out.println(list2.hashCode());  }  }  // What is the output hereQ1883 |
| |  | | --- | | A.  Content of the ArrayList |  |  | | --- | | B.  hashcode values of the ArrayList |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **205.** | package toString.pack3;  class D  {  int i;  int j;  D(int i, int j){  this.i = i;  this.j = j;  }  @Override  public int hashCode() {  String s1 = Integer.toString(i);  String s2 = Integer.toString(j);  int hash = s1.hashCode();  hash += s2.hashCode();  return hash;  }  }  public class M9 {  public static void main(String[] args) {  D d1 = new D(10, 20);  D d2 = new D(10, 20);  System.out.println(d1.hashCode());  System.out.println(d2.hashCode());  }  }  Q1884 |
|  | |
| **206.** | package toString.pack3;  class D  {  int i;  int j;  D(int i, int j){  this.i = i;  this.j = j;  }  @Override  public int hashCode() {  String s1 = Integer.toString(i);  String s2 = Integer.toString(j);  int hash = s1.hashCode();  hash += s2.hashCode();  return hash;  }  }  public class M9 {  public static void main(String[] args) {  D d1 = new D(10, 20);  D d2 = new D(10, 20);  System.out.println(d1.hashCode());  System.out.println(d2.hashCode());  D d3 = new D(20,10);  System.out.println(d3.hashCode());  D d4 = new D(200, 100);  System.out.println(d4.hashCode());  }  }  // What is the output hereQ1885 |
| |  | | --- | | A.  Content of the object |  |  | | --- | | B.  hashcode values of the object |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **207.** | package toString.pack4;  class B  {    }  public class M2 {  public static void main(String[] args) {  B b1 = new B();  B b2 = b1.clone();  }  }  // Will it compiles succesfullyQ1901 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |
| **208.** | package toString.pack4;  public class C {  public static void main(String[] args) {  C c1 = new C();  C c2 = c1.clone();  }  }  // will it compiles successfully or notQ1902 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |

|  |  |
| --- | --- |
| **209.** | package toString.pack4;  public class D {  public static void main(String[] args) {  D d1 = new D();  D d2 = (D) d1.clone();  System.out.println("done");  }  }  Q1903 |
| |  | | --- | | A.  done |  |  | | --- | | B.  Comilation Error |  |  | | --- | | C.  Runtime exception | | |
| **210.** | package toString.pack4;  public class E {  public static void main(String[] args)  throws CloneNotSupportedException  {  E e1 = new E();  System.out.println(1);  E e2 = (E) e1.clone();  System.out.println(2);  }    }  Q1904 |
| |  | | --- | | A.  1  2 |  |  | | --- | | B.  Comilation Error |  |  | | --- | | C.  1  CloneNotSupportedException | | |

|  |  |
| --- | --- |
| **211.** | package toString.pack4;  public class F implements Cloneable {  public static void main(String[] args)  throws CloneNotSupportedException{  F f1 = new F();  F f2 = (F) f1.clone();  System.out.println("f1: " + f1);  System.out.println("f2: " + f2);  }  }  // will it compiling and running successfullyQ1905 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |
| **212.** | package toString.pack4;  public class G implements Cloneable{    int i;  public static void main(String[] args)  throws CloneNotSupportedException  {  G obj1 = new G();  obj1.i = 10;  G obj2 = (G) obj1.clone();  System.out.println("a:" + obj2.i);  obj2.i = 20;  System.out.println("b:" + obj1.i);  obj1.i = 30;  System.out.println("c:" + obj2.i);  }  }  Q1906 |
| |  | | --- | | A.  a:10  b:10  c:20 |  |  | | --- | | B.  a:20  b:10  c:20 |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **213.** | package com.lara;  public class M2 {  public static void main(String[] args) {  for(int i = 1; i <= 1000; i++)  {  System.out.println("first loop :" + i);  }  for(int i = 2000; i <= 3000; i++)  {  System.out.println("second loop :" + i);  }  for(int i = 4000; i <= 5000; i++)  {  System.out.println("third loop :" + i);  }  }  }  /\* Guess what could be the output \*/Q2317 |
| |  | | --- | | A.  SequentialOutput |  |  | | --- | | B.  Simultaneous Output |  |  | | --- | | C.  Infinate Output |  |  | | --- | | D.  Compilation Error | | |
| **214.** | package com.lara;  class A extends Thread  {  @Override  public void run() {  for(int i = 2000; i <= 3000; i++)  {  System.out.println("second loop :" + i);  }  }  }  class B extends Thread  {  @Override  public void run() {  for(int i = 4000; i <= 5000; i++)  {  System.out.println("third loop :" + i);  }  }  }  public class M3 {  public static void main(String[] args) {  A a1 = new A();  a1.start();  B b1 = new B();  b1.start();  for(int i = 1; i <= 1000; i++)  {  System.out.println("first loop :" + i);  }  }  }  /\* Guess what could be the output \*/Q2318 |
| |  | | --- | | A.  SequentialOutput |  |  | | --- | | B.  Simultaneous Output |  |  | | --- | | C.  Infinate Output |  |  | | --- | | D.  Compilation Error | | |

|  |  |
| --- | --- |
| **215.** | package com.lara;  class C implements Runnable  {  @Override  public void run() {  for(int i = 2000; i <= 3000; i++)  {  System.out.println("second loop :" + i);  }  }  }  class D implements Runnable  {  @Override  public void run() {  for(int i = 4000; i <= 5000; i++)  {  System.out.println("third loop :" + i);  }  }  }  public class M4 {  public static void main(String[] args) {  C c1 = new C();  Thread t1 = new Thread(c1);  t1.start();    D d1 = new D();  Thread t2 = new Thread(d1);  t2.start();  for(int i = 1; i <= 1000; i++)  {  System.out.println("first loop :" + i);  }  }  }  Q2319 |
| |  | | --- | | A.  SequentialOutput |  |  | | --- | | B.  Simultaneous Output |  |  | | --- | | C.  Infinate Output |  |  | | --- | | D.  Compilation Error | | |
| **216.** | package com.lara;  class E extends Thread  {  @Override  public void run() {  for(int i = 1; i <= 1000; i++)  {  System.out.println(getName() + ":" + i);  }  }  }  public class M5 {  public static void main(String[] args) {  E e1 = new E();  e1.start();  //e1.start();    E e2 = new E();  e2.start();  for(int i = 5000; i <= 6000; i++)  {  System.out.println(Thread.currentThread().getName() + ":" + i);  }  }  }  Q2320 |
| |  | | --- | | A.  SequentialOutput |  |  | | --- | | B.  Simultaneous Output |  |  | | --- | | C.  Infinate Output |  |  | | --- | | D.  Compilation Error | | |

|  |  |
| --- | --- |
| **217.** | package com.lara;  class F implements Runnable  {  @Override  public void run() {  for(int i = 1; i <= 1000; i++)  {  System.out.println(Thread.currentThread().getName() + ":" + i);  }  }  }  public class M6 {  public static void main(String[] args) {  F f1 = new F();  Thread t1 = new Thread(f1);  t1.start();  //t1.start();    Thread t2 = new Thread(f1);  t2.start();  for(int i = 5000; i <= 6000; i++)  {  System.out.println(Thread.currentThread().getName() + ":" + i);  }  }  }  Q2321 |
| |  | | --- | | A.  SequentialOutput |  |  | | --- | | B.  Simultaneous Output |  |  | | --- | | C.  Infinate Output |  |  | | --- | | D.  Compilation Error | | |
| **218.** | package com.lara;  class G extends Thread  {  @Override  public void run() {  for(int i = 1; i <= 1000; i++)  {  System.out.println(Thread.currentThread().getName() + ":" + i);  }  }  }  public class M7 {  public static void main(String[] args) {  G g1 = new G();  //g1.start();  g1.run();  for(int i = 2000; i <= 3000; i++)  {  System.out.println(Thread.currentThread().getName() + ":" + i);  }    }  }  Q2322 |
| |  | | --- | | A.  SequentialOutput |  |  | | --- | | B.  Simultaneous Output |  |  | | --- | | C.  Infinate Output |  |  | | --- | | D.  Compilation Error | | |

|  |  |
| --- | --- |
| **219.** | package com.lara;  class H implements Runnable  {  @Override  public void run() {  for(int i = 1; i <= 1000; i++)  {  System.out.println(Thread.currentThread().getName() + ":" + i);  }  }  }  public class M8 {  public static void main(String[] args) {  H h1 = new H();  Thread t1 = new Thread(h1);  t1.start();  //h1.run();  //t1.run();  for(int i = 2000; i <= 3000; i++)  {  System.out.println(Thread.currentThread().getName() + ":" + i);  }    }  }  Q2323 |
| |  | | --- | | A.  SequentialOutput |  |  | | --- | | B.  Simultaneous Output |  |  | | --- | | C.  Infinate Output |  |  | | --- | | D.  Compilation Error | | |
| **220.** | package com.lara;  public class M9 {  public static void main(String[] args) {  Thread t1 = Thread.currentThread();  System.out.println(t1.getName());  System.out.println(t1.isDaemon());  System.out.println(t1.getPriority());  System.out.println(t1.getId());  }  }  Q2324 |
| |  | | --- | | A.  main  false  5  1 |  |  | | --- | | B.  main  true  5  1 |  |  | | --- | | C.  Compile Time Error |  |  | | --- | | D.  None |  |  | | --- | | E.  Runtime Error | | |

|  |  |
| --- | --- |
| **221.** | package com.lara;  class J extends Thread  {  @Override  public void run() {  System.out.println(getName());  System.out.println(isDaemon());  System.out.println(getPriority());  System.out.println(getId());  }  }  public class M10 {  public static void main(String[] args) {  J ref = new J();  ref.start();  }  }  Q2325 |
| |  | | --- | | A.  Thread-0  true  5  10 |  |  | | --- | | B.  Thread-0  false  5  10 |  |  | | --- | | C.  Compile Time Error |  |  | | --- | | D.  None |  |  | | --- | | E.  Runtime Error | | |
| **222.** | package com.lara;  public class M11 {  public static void main(String[] args) {  Thread t1 = Thread.currentThread();  System.out.println(t1.getName());  t1.setName("initiator");  System.out.println(t1.getName());  }  }  Q2326 |
| |  | | --- | | A.  initiator  main |  |  | | --- | | B.  Thread  initiator |  |  | | --- | | C.  main  initiator |  |  | | --- | | D.  Compilation Error |  |  | | --- | | E.  None | | |

|  |  |
| --- | --- |
| **223.** | package com.lara;  public class M12 {  public static void main(String[] args) {  Thread t1 = Thread.currentThread();  System.out.println(t1.isDaemon());  t1.setDaemon(true);  System.out.println(t1.isDaemon());  }  }  Q2327 |
| |  | | --- | | A.  false  true |  |  | | --- | | B.  true  false |  |  | | --- | | C.  Compile Time Error |  |  | | --- | | D.  RunTime Exception |  |  | | --- | | E.  None | | |
| **224.** | package com.lara;  public class M13 {  public static void main(String[] args) {  Thread t1 = Thread.currentThread();  System.out.println(t1.getPriority());  t1.setPriority(11);  System.out.println(t1.getPriority());  }  }  Q2328 |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  5  Exception |  |  | | --- | | C.  5  11 |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **225.** | package com.lara;  class K extends Thread  {  @Override  public void run() {  System.out.println("child thread");  }  }  public class M14 {  public static void main(String[] args) {  K k1 = new K();  k1.start();  System.out.println(k1.getName());  k1.setName("first child");  System.out.println(k1.getName());  }  }  Q2329 |
| |  | | --- | | A.  Thread-0  first child  child thread |  |  | | --- | | B.  Thread-0  child thread  first child |  |  | | --- | | C.  Compile Time Error | | |
| **226.** | package com.lara;  class L implements Runnable  {  @Override  public void run() {  System.out.println("child thread");  }  }  public class M15 {  public static void main(String[] args) {  L obj = new L();  Thread t1 = new Thread(obj);  t1.start();  System.out.println(t1.getName());  t1.setName("first child");  System.out.println(t1.getName());  }  }Q2330 |
| |  | | --- | | B.  Thread-0  first child  child thread |  |  | | --- | | B.  Thread-0  child thread  first child |  |  | | --- | | C.  Compile Time Error | | |

|  |  |
| --- | --- |
| **227.** | package com.lara;  class M extends Thread  {  M(String name)  {  super(name);  }  @Override  public void run() {  System.out.println("child thread");  }  }  public class M16 {  public static void main(String[] args) {  M m1 = new M("my first thread");  m1.start();  System.out.println(m1.getName());  m1.setName("first child");  System.out.println(m1.getName());  }  }Q2331 |
| |  | | --- | | A.  my first thread  child thread  first child |  |  | | --- | | B.  my first thread  first child  child thread |  |  | | --- | | C.  Compile Time Error |  |  | | --- | | D.  RunTime Exception | | |
| **228.** | package com.lara;  class N implements Runnable  {  @Override  public void run() {  System.out.println("child thread");  }  }  public class M17 {  public static void main(String[] args) {  N obj = new N();  Thread t1 = new Thread(obj, "my first thread");  t1.start();  System.out.println(t1.getName());  t1.setName("first child");  System.out.println(t1.getName());  }  }Q2332 |
| |  | | --- | | A.  my first thread  child thread  first child |  |  | | --- | | B.  my first thread  first child  child thread |  |  | | --- | | C.  Compile Time Error |  |  | | --- | | D.  RunTime Exception | | |

|  |  |
| --- | --- |
| **229.** | package com.lara;  class O extends Thread  {  @Override  public void run() {  System.out.println("from o thread:" + getPriority());  }  }  class P extends Thread  {  @Override  public void run() {  System.out.println("from p thread:" + getPriority());  setPriority(7);  O th1 = new O();  th1.start();  }  }  public class M18 {  public static void main(String[] args) {  Thread.currentThread().setPriority(10);  P p1 = new P();  p1.start();  }  }  Q2333 |
| |  | | --- | | A.  from o thread:7  from p thread:10 |  |  | | --- | | B.  from p thread:10  from o thread:7 |  |  | | --- | | C.  Compile Time Error |  |  | | --- | | D.  RunTime Exception | | |
| **230.** | package com.lara;  class Q extends Thread  {  @Override  public void run() {  for(int i = 0; i < 1000; i++)  {  System.out.println(i);  }  }  }  public class M19 {  public static void main(String[] args) {  Q q1 = new Q();  q1.start();  System.out.println("done");  }  }  /\* Guess what could be the output\*/Q2334 |
| |  | | --- | | A.  SequentialOutput |  |  | | --- | | B.  Simultaneous Output |  |  | | --- | | C.  Infinate Output |  |  | | --- | | D.  Compilation Error | | |

|  |  |
| --- | --- |
| **231.** | package com.lara;  class R extends Thread  {  @Override  public void run() {  for(int i = 0; i < 1000; i++)  {  System.out.println(i);  }  }  }  public class M20 {  public static void main(String[] args) {  R r1 = new R();  r1.setDaemon(true);  r1.start();  System.out.println("done");  }  }  Q2335 |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  Runtime Exception |  |  | | --- | | C.  done |  |  | | --- | | D.  None | | |
| **232.** | package com.lara;  class S extends Thread  {  @Override  public void run() {  for(int i = 0; i < 1000; i++)  {  System.out.println(i);  }  }  }  public class M21 {  public static void main(String[] args) {  S s1 = new S();  s1.start();  try  {  s1.join();  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  System.out.println("done");  }  }  /\* Guess What could be the output\*/Q2336 |
| |  | | --- | | A.  SequentialOutput |  |  | | --- | | B.  Simultaneous Output |  |  | | --- | | C.  Infinate Output |  |  | | --- | | D.  Compilation Error | | |

|  |  |
| --- | --- |
| **233.** | package com.lara;  public class M22 {  public static void main(String[] args) {  System.out.println("main begin");  for(int i = 1; i <= 10; i++)  {  System.out.println(i);  try  {  Thread.sleep(1000);  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  }  System.out.println("main end");  }  }  Q2337 |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  Runtime Exception |  |  | | --- | | C.  main begin  1  2  3  4  5  6  7  8  9  10  main end |  |  | | --- | | D.  1  main begin  2  3  4  5  6  7  8  9  10  main end | | |
| **234.** | package com.lara;  public class M23 {  public static void main(String[] args) {  System.out.println("main begin");  for(int i = 1; i <= 10; i++)  {  System.out.println(i);  try  {  Thread.sleep(2000, 500);  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  }  System.out.println("main end");  }  }  Q2338 |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  Runtime Exception |  |  | | --- | | C.  main begin  1  2  3  4  5  6  7  8  9  main end  10 |  |  | | --- | | D.  1  main begin  2  3  4  5  6  7  8  9  10  main end |  |  | | --- | | D.  main begin  1  2  3  4  5  6  7  8  9  10  main end | | |

|  |  |
| --- | --- |
| **235.** | package com.lara;  class T extends Thread  {  @Override  public void run() {  System.out.println("run begin");  for(int i = 100; i <= 110; i++)  {  System.out.println(i);  try  {  sleep(5000, 500);  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  }  System.out.println("run end");  }  }  public class M24 {  public static void main(String[] args) {  T obj = new T();  obj.start();  System.out.println("main begin");  for(int i = 1; i <= 10; i++)  {  System.out.println(i);  try  {  Thread.sleep(2000, 500);  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  }  System.out.println("main end");  }  }  Q2339 |
| |  | | --- | | A.  main begin  1  run begin  100  2  3  101  4  5  102  6  7  8  103  9  10  104  main end  105  106  107  108  109  110  run end |  |  | | --- | | B.  main begin  run begin  1  100  2  3  101  4  5  102  6  7  8  103  9  10  104  105  106  107  108  109  110  run end  main end |  |  | | --- | | C.  Compile Time Error |  |  | | --- | | D.  RunTime Exception | | |
| **236.** | package com.lara;  class U extends Thread  {  @Override  public void run() {  System.out.println("run begin");  for(int i = 100; i <= 110; i++)  {  System.out.println(i);  Util.sleep(5000);  }  System.out.println("run end");  }  }  public class M25 {  public static void main(String[] args) {  U obj = new U();  obj.start();  System.out.println("main begin");  for(int i = 1; i <= 10; i++)  {  System.out.println(i);  Util.sleep(2000);  }  System.out.println("main end");  }  }  class Util {  public static void sleep(long millis)  {  try  {  Thread.sleep(millis);  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  }  }  Q2340 |
| |  | | --- | | A.  main begin  run begin  1  100  2  3  101  4  5  102  6  7  8  103  9  10  104  105  106  107  108  109  110  run end  main end |  |  | | --- | | B.  main begin  1  run begin  100  2  3  101  4  5  102  6  7  8  103  9  10  104  main end  run end  105  106  107  108  109  110 |  |  | | --- | | C.  main begin  1  run begin  100  2  3  101  4  5  102  6  7  8  103  9  10  104  main end  105  106  107  108  109  110  run end |  |  | | --- | | D.  Compilation Error |  |  | | --- | | E.  Runtime Error | | |

|  |  |
| --- | --- |
| **237.** | package com.lara;  class V extends Thread  {  @Override  public void run() {  System.out.println("run begin");  Util.sleep(10000);  System.out.println("run end");  }  }  public class M26 {  public static void main(String[] args) {  System.out.println("main begin");  V obj = new V();  obj.start();  Util.sleep(2000);  obj.interrupt();  System.out.println("main end");  }  }  class Util {  public static void sleep(long millis)  {  try  {  Thread.sleep(millis);  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  }  }Q2341 |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  Runtime Exception | | |
| **238.** | package com.lara;  public class M27 {  public static void main(String[] args) {  System.out.println(1);  int i = 10 / 0;  System.out.println(2);  }  }  Q2342 |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  Runtime Exception |  |  | | --- | | C.  1  2 | | |

|  |  |
| --- | --- |
| **239.** | package com.lara;  class W extends Thread  {  @Override  public void run() {  System.out.println("run begin");  int i = 10 / 0;  System.out.println("run end");  }  }  public class M28 {  public static void main(String[] args) {  System.out.println(1);  W obj = new W();  obj.start();  System.out.println(2);  }  }  Q2343 |
| |  | | --- | | A.  1  2  run begin  Exception |  |  | | --- | | B.  1  run begin  run end  2 |  |  | | --- | | C.  run begin  run end  1  2 |  |  | | --- | | D.  Compilation Error | | |
| **240.** | package com.lara;  class X extends Thread  {  X()  {  start();  }  @Override  public void run() {  System.out.println("run begin");  System.out.println("run end");  }  }  public class M29 {  public static void main(String[] args) {  System.out.println(1);  X obj = new X();  System.out.println(2);  }  }  Q2344 |
| |  | | --- | | A.  1  2  run begin  run end |  |  | | --- | | B.  1  run begin  run end  2 |  |  | | --- | | C.  run begin  run end  1  2 |  |  | | --- | | D.  Compilation Error |  |  | | --- | | E.  Runtime Error | | |

|  |  |
| --- | --- |
| **241.** | package com.lara;  class Y extends Thread  {  Y()  {  start();  }  @Override  public void run() {  System.out.println("run begin");  System.out.println("run end");  }  }  public class M30 {  public static void main(String[] args) {  System.out.println(1);  Y obj = new Y();  obj.start();  System.out.println(2);  }  }  Q2345 |
| |  | | --- | | A.  1  run begin  run end  2 |  |  | | --- | | B.  1  run begin  run end  Exception |  |  | | --- | | C.  Compile Time Error | | |
| **242.** | package com.lara;  class Z extends Thread  {  Z()  {  start();  }  @Override  public void run() {  System.out.println("run begin");  start();  System.out.println("run end");  }  }  public class M31 {  public static void main(String[] args) {  System.out.println(1);  Z obj = new Z();  System.out.println(2);  }  }  Q2346 |
| |  | | --- | | A.  1  2  run begin  run end |  |  | | --- | | B.  1  run begin  run end  2 |  |  | | --- | | C.  run begin  run end  1  2 |  |  | | --- | | D.  1  2  run begin  Exception | | |

|  |  |
| --- | --- |
| **243.** | package com.lara.pack2;  class Test  {  int i;  }  class Util  {  static void sleep(long millis)  {  try  {  Thread.sleep(millis);  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  }  }  class A extends Thread  {  Test t1;  A(Test t1)  {  this.t1 = t1;  }  @Override  public void run() {  System.out.println("a:" + t1.i);  t1.i = 10;  Util.sleep(500);  System.out.println("b:" + t1.i);  t1.i = 20;  Util.sleep(500);  System.out.println("c:" + t1.i);  t1.i = 30;  }  }  class B extends Thread  {  Test t1;  B(Test t1)  {  this.t1 = t1;  }  @Override  public void run() {  System.out.println("d:" + t1.i);  t1.i = 40;  Util.sleep(500);  System.out.println("e:" + t1.i);  t1.i = 50;  Util.sleep(500);  System.out.println("f:" + t1.i);  t1.i = 60;  }  }  public class M1 {  public static void main(String[] args) {  Test t1 = new Test();  t1.i = 70;  A a1 = new A(t1);  a1.start();  Util.sleep(250);  B b1 = new B(t1);  b1.start();  Util.sleep(20000);  System.out.println("g:" + t1.i);  }  }  Q2377 |
| |  | | --- | | A.  a:70  d:10  b:40  e:20  c:50  f:30  g:60 |  |  | | --- | | B.  a:70  b:40  d:10  e:20  g:60  c:50  f:30 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | |
| **244.** | package com.lara.pack3;  class Util  {  static void sleep(long millis)  {  try  {  Thread.sleep(millis);  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  }  }  class A extends Thread  {  ThreadLocal t1;  A(ThreadLocal t1)  {  this.t1 = t1;  }  @Override  public void run() {  System.out.println("a:" + t1.get());  t1.set(10);  Util.sleep(500);  System.out.println("b:" + t1.get());  t1.set(20);  Util.sleep(500);  System.out.println("c:" + t1.get());  t1.set(30);  }  }  class B extends Thread  {  ThreadLocal t1;  B(ThreadLocal t1)  {  this.t1 = t1;  }  @Override  public void run() {  System.out.println("d:" + t1.get());  t1.set(40);  Util.sleep(500);  System.out.println("e:" + t1.get());  t1.set(50);  Util.sleep(500);  System.out.println("f:" + t1.get());  t1.set(60);  }  }  public class M1 {  public static void main(String[] args) {  ThreadLocal t1 = new ThreadLocal();  t1.set(70);  A a1 = new A(t1);  a1.start();  Util.sleep(250);  B b1 = new B(t1);  b1.start();  Util.sleep(20000);  System.out.println("g:" + t1.get());  }  }  Q2378 |
| |  | | --- | | A.  a:70  d:10  b:40  e:20  c:50  f:30  g:60 |  |  | | --- | | B.  a:null  d:null  b:10  e:40  c:20  f:50  g:70 |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **245.** | package com.lara.pack4;  class A extends Thread  {  A(ThreadGroup tg, String name)  {  super(tg, name);  }  @Override  public void run() {  for(int i = 1; i <= 1000; i++)  {  System.out.println(getName() + ":" + i);  }  }  }  class B extends Thread  {  B(ThreadGroup tg, String name)  {  super(tg, name);  }    @Override  public void run() {  for(int i = 1; i <= 1000; i++)  {  System.out.println(getName() + ":" + i);  }  }  }  class C implements Runnable  {  @Override  public void run() {  for(int i = 1; i <= 1000; i++)  {  System.out.println(Thread.currentThread().getName() + ":" + i);  }  }  }  public class M1 {  public static void main(String[] args) {  ThreadGroup tg = new ThreadGroup("myFirstGroup");  A a1 = new A(tg, "firstThread");  A a2 = new A(tg, "secondThread");  B b1 = new B(tg, "thirdThread");  B b2 = new B(tg, "fourthThread");    C c1 = new C();  Thread t1 = new Thread(tg, c1, "5thThread");  Thread t2 = new Thread(tg, c1, "6thThread");    a1.start();  a2.start();  b1.start();  b2.start();  t1.start();  t2.start();    tg.stop();  }  }  // Check which option is possible for this program  Q2379 |
| |  | | --- | | A.  We will get the output |  |  | | --- | | B.  we won't get output |  |  | | --- | | C.  Both a and b possible |  |  | | --- | | D.  None | | |
| **246.** | package com.lara.pack5;  class Employee  {  //several members  }  class A extends Thread  {  @Override  public void run() {  // TODO Auto-generated method stub  super.run();  }  }  class B extends Employee implements Runnable  {  @Override  public void run() {  // TODO Auto-generated method stub    }  }  public class M1 {  public static void main(String[] args) {  A a1 = new A();  a1.start();  A a2 = new A();  a2.start();  A a3 = new A();  a3.start();      B b1 = new B();  Thread t1 = new Thread(b1);  t1.start();    Thread t2 = new Thread(b1);  t2.start();    Thread t3 = new Thread(b1);  t3.start();                }  }  // check whether it compiles sucessful or not  Q2380 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |

|  |  |
| --- | --- |
| **247.** | package com.lara.pack1;  public class M1 {  public static void main(String[] args) {  Thread.State[] states = Thread.State.values();  for(Thread.State state : states)  {  System.out.println(state);  }  }  }  Q2381 |
| |  | | --- | | A.  Compile Time error |  |  | | --- | | B.  Runtime Error |  |  | | --- | | C.  NEW  RUNNABLE  BLOCKED  WAITING  TIMED\_WAITING  TERMINATED |  |  | | --- | | D.  NEW  RUNNABLE  BLOCKED  TERMINATED | | |
| **248.** | package com.lara.pack1;  class A extends Thread  {  @Override  public void run() {  for(int i = 1; i <= 1000; i++)  {  System.out.println(i);  }  }  }  public class M2 {  public static void main(String[] args) throws InterruptedException{  A a1 = new A();  System.out.println("a:" + a1.getState());  a1.start();  System.out.println("b:" + a1.getState());  Thread.sleep(7);  System.out.println("c:" + a1.getState());  Thread.sleep(10000);  System.out.println("d:" + a1.getState());  }  }  Q2382 |
| |  | | --- | | A.  SequentialOutput with thread states |  |  | | --- | | B.  Simultaneous Output with thread states |  |  | | --- | | C.  Infinate Output with thread states |  |  | | --- | | D.  Compilation Error | | |

|  |  |
| --- | --- |
| **249.** | package com.lara.pack1;  class B extends Thread  {  @Override  public void run() {  try  {  sleep(10000);  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  }  }  public class M3 {  public static void main(String[] args) throws InterruptedException{  B b1 = new B();  b1.start();  Thread.sleep(2000);  System.out.println(b1.getState());  }  }  Q2383 |
| |  | | --- | | A.  WAITING |  |  | | --- | | B.  BLOCKED |  |  | | --- | | C.  TIMED\_WAITING |  |  | | --- | | D.  RUNNABLE | | |
| **250.** | package com.lara.pack1;  class C extends Thread{  @Override  public void run() {  synchronized (this) {  try  {  wait();  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  }  }  }  public class M4 {  public static void main(String[] args) throws InterruptedException {  C c1 = new C();  c1.start();  Thread.sleep(100);  System.out.println(c1.getState());  }  }  Q2384 |
| |  | | --- | | A.  WAITING |  |  | | --- | | B.  RUNNING |  |  | | --- | | C.  RUNNABLE |  |  | | --- | | D.  TERMINATING | | |

|  |  |
| --- | --- |
| **251.** | package com.lara.pack1;  class D extends Thread {    Thread mainThread;    D(Thread mainThread)  {  this.mainThread = mainThread;  }    @Override  public void run() {  try  {  sleep(100);  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  System.out.println("state of main thread:" + mainThread.getState());  }  }  public class M5 {  public static void main(String[] args) {  Thread t1 = Thread.currentThread();  D d1 = new D(t1);  d1.start();  try  {  d1.join();  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }    }  }  Q2385 |
| |  | | --- | | A.  state of main thread:RUNNING |  |  | | --- | | B.  state of main thread:WAITING |  |  | | --- | | C.  state of main thread:TERMINATING |  |  | | --- | | D.  Compilation Error |  |  | | --- | | E.  Runtime Error | | |
| **252.** | package com.lara.pack1;  class E extends Thread  {  @Override  public void run() {  for(int i = 1; i <= 1000; i++)  {  System.out.println(i);  yield();  }  }  }  class F extends Thread  {  @Override  public void run() {  for(int i = 2000; i <= 3000; i++)  {  System.out.println(i);  yield();  }  }  }  public class M6 {  public static void main(String[] args) {  E e1 = new E();  e1.start();    F f1 = new F();  f1.start();  }  }  Q2386 |
| |  | | --- | | A.  SequentialOutput |  |  | | --- | | B.  Simultaneous Output |  |  | | --- | | C.  Infinate Output |  |  | | --- | | D.  Compilation Error | | |

|  |  |
| --- | --- |
| **253.** | package com.lara.pack1;  class Shared  {  synchronized void test1()  {  Thread t1 = Thread.currentThread();  for(int i = 1; i <= 1000; i++)  {  System.out.println("from test1 " + i + " by " + t1.getName());  }  }  synchronized void test2()  {  Thread t1 = Thread.currentThread();  for(int i = 1; i <= 1000; i++)  {  System.out.println("from test2 " + i + " by " + t1.getName());  }  }  }  class Thread1 extends Thread  {  Shared s1;  Thread1(Shared s1)  {  this.s1 = s1;  }  @Override  public void run() {  s1.test1();  }  }  class Thread2 extends Thread  {  Shared s1;  Thread2(Shared s1)  {  this.s1 = s1;  }  @Override  public void run() {  s1.test1();  }  }  public class M1 {  public static void main(String[] args) {  Shared s1 = new Shared();  Shared s2 = new Shared();    Thread1 t1 = new Thread1(s1);  Thread2 t2 = new Thread2(s2);    t1.start();  t2.start();  }  }  Q2387 |
| |  | | --- | | A.  Sequential output |  |  | | --- | | B.  simultaneous output |  |  | | --- | | C.  No output |  |  | | --- | | D.  Compilation Error | | |
| **254.** | package com.lara.pack2;  class Shared  {  void test1()  {  Thread t1 = Thread.currentThread();  synchronized(this)  {  for(int i = 1; i <= 1000; i++)  {  System.out.println("from test1 " + i + " by " + t1.getName());  }  }  }  void test2()  {  Thread t1 = Thread.currentThread();  synchronized(this)  {  for(int i = 1; i <= 1000; i++)  {  System.out.println("from test2 " + i + " by " + t1.getName());  }  }  }  }  class Thread1 extends Thread  {  Shared s1;  Thread1(Shared s1)  {  this.s1 = s1;  }  @Override  public void run() {  s1.test1();  }  }  class Thread2 extends Thread  {  Shared s1;  Thread2(Shared s1)  {  this.s1 = s1;  }  @Override  public void run() {  s1.test2();  }  }  public class M1 {  public static void main(String[] args) {  Shared s1 = new Shared();  Shared s2 = new Shared();    Thread1 t1 = new Thread1(s1);  Thread2 t2 = new Thread2(s2);    t1.start();  t2.start();  }  }  Q2388 |
| |  | | --- | | A.  Sequential output |  |  | | --- | | B.  simultaneous output |  |  | | --- | | C.  No output |  |  | | --- | | D.  Compilation Error | | |

|  |  |
| --- | --- |
| **255.** | package com.lara.pack3;  class A  {  static void test1()  {  Thread t1 = Thread.currentThread();  synchronized (A.class)  {  for(int i = 1; i <= 1000; i++)  {  System.out.println("from test1 with " + i + " by " + t1.getName());  }  }  }  static void test2()  {  Thread t1 = Thread.currentThread();  synchronized (A.class)  {  for(int i = 1; i <= 1000; i++)  {  System.out.println("from test2 with " + i + " by " + t1.getName());  }  }  }  }  class Thread1 extends Thread  {  @Override  public void run() {  A.test1();  }  }  class Thread2 extends Thread  {  @Override  public void run() {  A.test2();  }  }  public class M1 {  public static void main(String[] args) {  Thread1 t1 = new Thread1();  t1.start();    Thread2 t2 = new Thread2();  t2.start();  }  }  Q2389 |
| |  | | --- | | A.  Sequential output |  |  | | --- | | B.  simultaneous output |  |  | | --- | | C.  No output |  |  | | --- | | D.  Compilation Error | | |
| **256.** | package com.lara.pack4;  import java.lang.management.ManagementFactory;  import java.lang.management.ThreadMXBean;  import java.util.Arrays;  class Shared  {  synchronized void test1(Shared obj)  {  Thread t1 = Thread.currentThread();  System.out.println("test1 begin by " + t1.getName());  Util.sleep(1000);  obj.test2(this);  System.out.println("test1 end by " + t1.getName());  }  synchronized void test2(Shared obj)  {  Thread t1 = Thread.currentThread();  System.out.println("test2 begin by " + t1.getName());  Util.sleep(1000);  obj.test1(this);  System.out.println("test2 end by " + t1.getName());  }  }  class Util  {  static void sleep(long millis)  {  try  {  Thread.sleep(millis);  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  }  }  class Thread1 extends Thread  {  Shared s1, s2;  Thread1(Shared s1, Shared s2)  {  this.s1 = s1;  this.s2 = s2;  }  @Override  public void run() {  s1.test1(s2);  }  }  class Thread2 extends Thread  {  Shared s1, s2;  Thread2(Shared s1, Shared s2)  {  this.s1 = s1;  this.s2 = s2;  }  @Override  public void run() {  s2.test2(s1);  }  }  public class M1 {  public static void main(String[] args) {  Shared s1 = new Shared();  Shared s2 = new Shared();  Thread1 t1 = new Thread1(s1, s2);  t1.start();  Util.sleep(100);  Thread2 t2 = new Thread2(s1, s2);  t2.start();    Util.sleep(2000);    ThreadMXBean tmx = ManagementFactory.getThreadMXBean();  long[] ids = tmx.findDeadlockedThreads();    if(ids != null)  {  System.out.println("Threads are under dead lock");  System.out.println("dead locked thread ids: " + Arrays.toString(ids));  }  else  {  System.out.println("no threads are under dead lock");  }  System.out.println(t1.getState());  System.out.println(t2.getState());    }  }  Q2390 |
| |  | | --- | | A.  Sequential output |  |  | | --- | | B.  simultaneous output |  |  | | --- | | C.  No output |  |  | | --- | | D.  test1 begin by Thread-0  test2 begin by Thread-1  Threads are under dead lock  dead locked thread ids: [11, 10]  BLOCKED  BLOCKED |  |  | | --- | | E.  Compilation error | | |

|  |  |
| --- | --- |
| **257.** | package com.lara.pack5;  class A  {  synchronized void test1()  {  System.out.println("test1 on " + this + " begin by " + Thread.currentThread().getName());  try  {  wait();  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  System.out.println("test1 on " + this + " end by " + Thread.currentThread().getName());  }  synchronized void test2()  {  System.out.println("test2 on " + this + " begin by " + Thread.currentThread().getName());  notifyAll();  System.out.println("test2 on " + this + " end by " + Thread.currentThread().getName());  }  }  class Thread1 extends Thread  {  A obj;  Thread1(A obj)  {  this.obj = obj;  }  @Override  public void run() {  obj.test1();  }  }  class Thread2 extends Thread  {  A obj;  Thread2(A obj)  {  this.obj = obj;  }  @Override  public void run() {  obj.test1();  }  }  public class M1 {  public static void main(String[] args) throws InterruptedException{  A obj1 = new A();  A obj2 = new A();  Thread1 t1 = new Thread1(obj1);  t1.start();  Thread2 t2 = new Thread2(obj1);  t2.start();  Thread.sleep(10000);  System.out.println("main after 10 sec sleep");  obj2.test2();  }  }  //Check whether this program threads under dead lock or not  Q2391 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |
| **258.** | package com.lara.pack6;  class A  {  synchronized void test1()  {  System.out.println("test1 on " + this + " begin by " + Thread.currentThread().getName());  try  {  wait();  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  System.out.println("test1 on " + this + " end by " + Thread.currentThread().getName());  }  }  class Thread1 extends Thread  {  A obj;  Thread1(A obj)  {  this.obj = obj;  }  @Override  public void run() {  obj.test1();  }  }  class Thread2 extends Thread  {  A obj;  Thread2(A obj)  {  this.obj = obj;  }  @Override  public void run() {  obj.test1();  }  }  public class M1 {  public static void main(String[] args) throws InterruptedException{  A obj1 = new A();  A obj2 = new A();  Thread1 t1 = new Thread1(obj1);  t1.start();  Thread2 t2 = new Thread2(obj1);  t2.start();  Thread.sleep(10000);  System.out.println("main after 10 sec sleep");  synchronized (obj1)  {  System.out.println("test2 on " + obj1 + " begin by " + Thread.currentThread().getName());  obj1.notifyAll();  System.out.println("test2 on " + obj1 + " end by " + Thread.currentThread().getName());  }  }  }  Q2392 |
| |  | | --- | | A.  threads under dead lock |  |  | | --- | | B.  No dead lock |  |  | | --- | | C.  test1 on th.A@2d4021ce begin by Thread-0  test1 on th.A@2d4021ce begin by Thread-1  main after 10 sec sleep  test2 on th.A@2d4021ce begin by main  test2 on th.A@2d4021ce end by main  test1 on th.A@2d4021ce end by Thread-1  test1 on th.A@2d4021ce end by Thread-0 |  |  | | --- | | D.  test1 on th.A@2d4021ce begin by Thread-0  test1 on th.A@2d4021ce begin by Thread-1  main after 10 sec sleep  test2 on th.A@2d4021ce begin by main  test2 on th.A@2d4021ce end by main  test1 on th.A@2d4021ce end by Thread-0  test1 on th.A@2d4021ce end by Thread-1 | | |

|  |  |
| --- | --- |
| **259.** | package com.lara.pack7;  class A  {  }  class Thread1 extends Thread  {  A obj;  Thread1(A obj)  {  this.obj = obj;  }  @Override  public void run() {    synchronized (obj)  {  System.out.println("wait on " + obj + " begin by " + Thread.currentThread().getName());  try  {  obj.wait();  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  System.out.println("wait on " + obj + " end by " + Thread.currentThread().getName());  }  }  }  class Thread2 extends Thread  {  A obj;  Thread2(A obj)  {  this.obj = obj;  }  @Override  public void run() {  synchronized (obj)  {  System.out.println("wait on " + obj + " begin by " + Thread.currentThread().getName());  try  {  obj.wait();  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  System.out.println("wait on " + obj + " end by " + Thread.currentThread().getName());  }  }  }  public class M1 {  public static void main(String[] args) throws InterruptedException{  A obj1 = new A();  A obj2 = new A();  Thread1 t1 = new Thread1(obj1);  t1.start();  Thread2 t2 = new Thread2(obj1);  t2.start();  Thread.sleep(10000);  System.out.println("main after 10 sec sleep");  synchronized (obj1)  {  System.out.println("test2 on " + obj1 + " begin by " + Thread.currentThread().getName());  obj1.notifyAll();  System.out.println("test2 on " + obj1 + " end by " + Thread.currentThread().getName());  }  }  }  Q2393 |
| |  | | --- | | A.  threads under waiting |  |  | | --- | | B.  No waiting |  |  | | --- | | C.  wait on th.A@64fe8248 begin by Thread-0  wait on th.A@64fe8248 begin by Thread-1  main after 10 sec sleep  test2 on th.A@64fe8248 begin by main  test2 on th.A@64fe8248 end by main  wait on th.A@64fe8248 end by Thread-0  wait on th.A@64fe8248 end by Thread-1 |  |  | | --- | | D.  wait on th.A@64fe8248 begin by Thread-0  wait on th.A@64fe8248 begin by Thread-1  main after 10 sec sleep  test2 on th.A@64fe8248 begin by main  test2 on th.A@64fe8248 end by main  wait on th.A@64fe8248 end by Thread-1  wait on th.A@64fe8248 end by Thread-0 | | |
| **260.** | package com.lara.pack8;  class Thread1 extends Thread  {  @Override  public void run() {    synchronized (this)  {  System.out.println("wait on " + this + " begin by " + Thread.currentThread().getName());  try  {  this.wait();  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  System.out.println("wait on " + this + " end by " + Thread.currentThread().getName());  }  }  }  class Thread2 extends Thread  {  public void run() {    synchronized (this)  {  System.out.println("wait on " + this + " begin by " + Thread.currentThread().getName());  try  {  this.wait();  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  System.out.println("wait on " + this + " end by " + Thread.currentThread().getName());  }  }  }  public class M1 {  public static void main(String[] args) throws InterruptedException{  Thread1 t1 = new Thread1();  t1.start();  Thread2 t2 = new Thread2();  t2.start();  Thread.sleep(10000);  System.out.println("main after 10 sec sleep");  synchronized (t1)  {  System.out.println("test2 on " + t1 + " begin by " + Thread.currentThread().getName());  t1.notify();  System.out.println("test2 on " + t1 + " end by " + Thread.currentThread().getName());  }  synchronized (t2)  {  System.out.println("test2 on " + t2 + " begin by " + Thread.currentThread().getName());  t2.notify();  System.out.println("test2 on " + t2 + " end by " + Thread.currentThread().getName());  }      }  }  Q2394 |
| |  | | --- | | A.  wait on Thread[Thread-0,5,main] begin by Thread-0  wait on Thread[Thread-1,5,main] begin by Thread-1  main after 10 sec sleep  test2 on Thread[Thread-0,5,main] begin by main  test2 on Thread[Thread-0,5,main] end by main  test2 on Thread[Thread-1,5,main] begin by main  wait on Thread[Thread-0,5,main] end by Thread-0  test2 on Thread[Thread-1,5,main] end by main  wait on Thread[Thread-1,5,main] end by Thread-1 |  |  | | --- | | B.  wait on Thread[Thread-0,5,main] begin by Thread-1  wait on Thread[Thread-1,5,main] begin by Thread-0  main after 10 sec sleep  test2 on Thread[Thread-0,5,main] begin by main  test2 on Thread[Thread-0,5,main] end by main  test2 on Thread[Thread-1,5,main] begin by main  wait on Thread[Thread-0,5,main] end by Thread-1  test2 on Thread[Thread-1,5,main] end by main  wait on Thread[Thread-1,5,main] end by Thread-0 |  |  | | --- | | C.  wait on Thread[Thread-0,5,main] begin by Thread-0  wait on Thread[Thread-1,5,main] begin by Thread-1  main after 10 sec sleep  wait on Thread[Thread-0,5,main] end by Thread-0  test2 on Thread[Thread-1,5,main] end by main  wait on Thread[Thread-1,5,main] end by Thread-1 |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **261.** | package com.lara.pack9;  class Thread1 extends Thread  {  @Override  public void run() {    synchronized (this)  {  System.out.println("wait on " + this + " begin by " + Thread.currentThread().getName());  try  {  this.wait();  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  System.out.println("wait on " + this + " end by " + Thread.currentThread().getName());  }  }  }  class Thread2 extends Thread  {  Thread1 obj;  Thread2(Thread1 obj)  {  this.obj = obj;  }  public void run() {    synchronized (obj)  {  System.out.println("notify on " + obj + " begin by " + Thread.currentThread().getName());  obj.notify();  System.out.println("notify on " + obj + " end by " + Thread.currentThread().getName());  }  }  }  public class M1 {  public static void main(String[] args) throws InterruptedException{  Thread1 t1 = new Thread1();  t1.start();  Thread.sleep(10000);  System.out.println("main after 10 sec sleep");  Thread2 t2 = new Thread2(t1);  t2.start();  }  }  Q2395 |
| |  | | --- | | A.  wait on Thread[Thread-0,5,main] begin by Thread-1  main after 10 sec sleep  notify on Thread[Thread-0,5,main] begin by Thread-0  notify on Thread[Thread-0,5,main] end by Thread-0  wait on Thread[Thread-0,5,main] end by Thread-1 |  |  | | --- | | B.  wait on Thread[Thread-0,5,main] begin by Thread-0  main after 10 sec sleep  notify on Thread[Thread-0,5,main] begin by Thread-1  notify on Thread[Thread-0,5,main] end by Thread-1  wait on Thread[Thread-0,5,main] end by Thread-0 |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |
| **262.** | package com.lara.pack10;  import java.util.ArrayList;  class Util  {  static void sleep(long millis)  {  try  {  Thread.sleep(millis);  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  }  }  class CommonTaskThread extends Thread  {  @Override  public void run()  {  while(true)  {  //wait  synchronized (this) {  try  {  wait();  }  catch(InterruptedException ex)  {  ex.printStackTrace();  }  }    //common task. it can be any  for(int i = 1; i <= 10; i++)  {  System.out.println(i + " by " + getName());  Util.sleep(1000);  }    //notify  synchronized (this) {  notify();  }  }  }  }  class ThreadPoolManager  {  private ArrayList<CommonTaskThread> pool = new ArrayList<CommonTaskThread>();    public void init()  {  CommonTaskThread ct = null;  for(int i = 1; i <= 10; i++)  {  ct = new CommonTaskThread();  ct.start();  pool.add(ct);  }  }    public CommonTaskThread checkOut()  {  CommonTaskThread ct = null;  if(pool.size() > 0)  {  ct = pool.remove(0);  }  else  {  ct = new CommonTaskThread();  ct.start();  }  return ct;  }    public void checkIn(CommonTaskThread ct)  {  if(pool.size() < 10)  {  pool.add(ct);  }  else  {  ct = null;  }  }    public void release()  {  CommonTaskThread ct = null;  for(int i = 0; i < pool.size(); )  {  ct = pool.remove(0);  ct = null;  }  pool = null;  }  }  class Customer1 extends Thread  {  private ThreadPoolManager tpm;  Customer1(ThreadPoolManager tpm)  {  this.tpm = tpm;  }    @Override  public void run() {  while(true)  {  System.out.println("customer1 is trying to get a thread from the pool");  CommonTaskThread ct = tpm.checkOut();  synchronized (ct) {  ct.notify();  }  synchronized (ct) {  try {  ct.wait();  }  catch(InterruptedException ex) {  ex.printStackTrace();  }  }  System.out.println("customer1 is sending used thread back to the pool");  tpm.checkIn(ct);  Util.sleep(10000);  }  }  }  class Customer2 extends Thread  {  private ThreadPoolManager tpm;  Customer2(ThreadPoolManager tpm)  {  this.tpm = tpm;  }    @Override  public void run() {  while(true)  {  System.out.println("customer2 is trying to get a thread from the pool");  CommonTaskThread ct = tpm.checkOut();  synchronized (ct) {  ct.notify();  }  synchronized (ct) {  try {  ct.wait();  }  catch(InterruptedException ex) {  ex.printStackTrace();  }  }  System.out.println("customer2 is sending used thread back to the pool");  tpm.checkIn(ct);  Util.sleep(10000);  }  }  }  class Customer3 extends Thread  {  private ThreadPoolManager tpm;  Customer3(ThreadPoolManager tpm)  {  this.tpm = tpm;  }    @Override  public void run() {  while(true)  {  System.out.println("customer3 is trying to get a thread from the pool");  CommonTaskThread ct = tpm.checkOut();  synchronized (ct) {  ct.notify();  }  synchronized (ct) {  try {  ct.wait();  }  catch(InterruptedException ex) {  ex.printStackTrace();  }  }  System.out.println("customer3 is sending used thread back to the pool");  tpm.checkIn(ct);  Util.sleep(10000);  }  }  }  public class M1 {  public static void main(String[] args) {  ThreadPoolManager tpm = new ThreadPoolManager();  tpm.init();    Customer1 c1 = new Customer1(tpm);  c1.start();    Customer2 c2 = new Customer2(tpm);  c2.start();    Customer3 c3 = new Customer3(tpm);  c3.start();    Util.sleep(1000 \* 1000);  c1.stop();  c2.stop();  c3.stop();  Util.sleep(20000);  tpm.release();  System.out.println("end of the game");  }  }  Q2396 |
| |  | | --- | | A.  Sequential output |  |  | | --- | | B.  Simultaneous output |  |  | | --- | | C.  infinite output |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **263.** | Lambda expression was introduced in which version of jdk?Q2854 |
| |  | | --- | | A.  jdk 1.5 |  |  | | --- | | B.  jk 1.8 |  |  | | --- | | C.  jdk 8.0 |  |  | | --- | | D.  none of the above | | |
| **264.** | lambda expression was used only for what?Q2855 |
| |  | | --- | | A.  sinle metod interface |  |  | | --- | | B.  multi level interface |  |  | | --- | | C.  jdk 8.0 |  |  | | --- | | D.  none of the above | | |

|  |  |
| --- | --- |
| **265.** | single method interface is also known as what?Q2856 |
| |  | | --- | | A.  method interface |  |  | | --- | | B.  attribute interface |  |  | | --- | | C.  functional intreface |  |  | | --- | | D.  none of the above | | |
| **266.** | If the inner class was declared as static then that is known as what?Q2869 |
| |  | | --- | | A.  non static inner class |  |  | | --- | | B.  static inner class |  |  | | --- | | C.  global inner class |  |  | | --- | | D.  local inner class | | |

|  |  |
| --- | --- |
| **267.** | If the inner class was declared as non-static then that is known as what?Q2870 |
| |  | | --- | | A.  non static inner class |  |  | | --- | | B.  static inner class |  |  | | --- | | C.  global inner class |  |  | | --- | | D.  local inner class | | |
| **268.** | can a outer class declered as static?Q2871 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no |  |  | | --- | | C.  depends on code | | |

|  |  |
| --- | --- |
| **269.** | can a inner class declered as static or non static?Q2872 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no |  |  | | --- | | C.  depends on code | | |
| **270.** | can we use a non static member inside a static member without reference variable?Q2873 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **271.** | to use a non static member insidde a static member which variable we need?Q2874 |
| |  | | --- | | A.  datattype variable |  |  | | --- | | B.  local variable |  |  | | --- | | C.  global variable |  |  | | --- | | D.  reference variable | | |
| **272.** | class A{  class B{}  static class C{}  public static void main(String[] args){  B b1=new B();  C c1=new C();  System.out.println("done");}}  what is the output?    Q2875 |
| |  | | --- | | A.  compilation error |  |  | | --- | | B.  done |  |  | | --- | | C.  error |  |  | | --- | | D.  none of the above | | |

|  |  |
| --- | --- |
| **273.** | class A{  class B{}  static class C{}  public static void main(String[] args){  B b1=null;  C c1= null;  System.out.println("done");}}  what is the output?    Q2876 |
| |  | | --- | | A.  compilation error |  |  | | --- | | B.  done |  |  | | --- | | C.  error |  |  | | --- | | D.  none of the above | | |
| **274.** | the class which will be kept along with the main method that is known as what?Q2877 |
| |  | | --- | | A.  inner class |  |  | | --- | | B.  static inner class |  |  | | --- | | C.  local inner class |  |  | | --- | | D.  non staic inner class | | |

|  |  |
| --- | --- |
| **275.** | To return one class object to the main class which method user need to call?Q2878 |
| |  | | --- | | A.  get() |  |  | | --- | | B.  class() |  |  | | --- | | C.  new class() |  |  | | --- | | D.  getclass() | | |
| **276.** | what is available in the class object that the user gets by calling getclass() ?Q2879 |
| |  | | --- | | A.  object |  |  | | --- | | B.  method |  |  | | --- | | C.  getter() |  |  | | --- | | D.  setter() | | |

|  |  |
| --- | --- |
| **277.** | In which version of java static import concept was introduced?Q2880 |
| |  | | --- | | A.  jdk 8.0 |  |  | | --- | | B.  jdk 11.2 |  |  | | --- | | C.  jdk1.5 |  |  | | --- | | D.  none of the above | | |
| **278.** | to accsess which members of a package to another package is known as static import?Q2881 |
| |  | | --- | | A.  non static |  |  | | --- | | B.  derived |  |  | | --- | | C.  inherited |  |  | | --- | | D.  static | | |

|  |  |
| --- | --- |
| **279.** | to accsess static members of a package to another package is known as what?Q2882 |
| |  | | --- | | A.  package import |  |  | | --- | | B.  static import |  |  | | --- | | C.  interface import |  |  | | --- | | D.  none of the above | | |
| **280.** | by static import which one we cant accsess of .java file?Q2883 |
| |  | | --- | | A.  static member |  |  | | --- | | B.  class |  |  | | --- | | C.  interface import |  |  | | --- | | D.  enum | | |

|  |  |
| --- | --- |
| **281.** | Class inside another class is known as what?Q2889 |
| |  | | --- | | A.  class |  |  | | --- | | B.  nested class |  |  | | --- | | C.  inner class |  |  | | --- | | D.  none of the avobe | | |
| **282.** | Which is true about an anonymous inner class?Q2890 |
| |  | | --- | | A.  It can extend exactly one class and implement exactly one interface. |  |  | | --- | | B.  It can extend exactly one class and can implement multiple interfaces. |  |  | | --- | | C.  It can extend exactly one class or implement exactly one interface. |  |  | | --- | | D.  It can implement multiple interfaces regardless of whether it also extends a class. | | |

|  |  |
| --- | --- |
| **283.** | Which is true about a method-local inner class?Q2891 |
| |  | | --- | | A.  It must be marked final. |  |  | | --- | | B.  It can be marked abstract. |  |  | | --- | | C.  It can be marked public. |  |  | | --- | | D.  It can be marked static. | | |

[1](https://lara.co.in/#qid27)23456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104105106107108109110111112113114115116117118119120121122123124125126127128129130131132133134135136137138139140141142143144145146147148149150151152153154155156157158159160161162163164165166167168169170171172173174175176177178179180181182183184185186187188189190191192193194195196197198199200201202203204205206207208209210211212213214215216217218219220221222223224225226227228229230231232233234235236237238239240241242243244245246247248249250251252253254255256257258259260261262263264265266267268269270271272273274275276277278279280281282