**Core Java Advanced Part – 3**

  Strings

  basics and important methods (Total available questions : 25)

  StringBuffer (Total available questions : 13)

  Formatters (Total available questions : 16)

  Regular expressions (Total available questions : 27)

  DateFormaters and NumberFormatters (Total available questions : 15)

  Arrays

  arrays (Total available questions : 10)

  Collection API

  introduction (Total available questions : 6)

  basic operations (Total available questions : 30)

  different collections (Total available questions : 15)

  list (Total available questions : 27)

  queues (Total available questions : 27)

  avoiding duplicates (set) (Total available questions : 27)

  maps (Total available questions : 44)

  iterators (Total available questions : 69)

  sorting (Total available questions : 35)

  TreeSet and TreeMap (Total available questions : 6)

  sorting map based on values (Total available questions : 17)

  modifying collections classes as synchronized (Total available questions : 13)

  concurrent package

  sort even and odd separately

|  |  |
| --- | --- |
| **1.** | is it possible to read the elements of ArrayList through for-each loopQ48 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **2.** | is it possible to read the elements of ArrayList through IteratorQ49 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **3.** | is it possible to read the elements of ArrayList through ListIteratorQ50 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **4.** | What is the return type of hasNext() method in the Iterator?Q51 |
| |  | | --- | | A.  double |  |  | | --- | | B.  int |  |  | | --- | | C.  boolean |  |  | | --- | | D.  String |  |  | | --- | | E.  Object | | |

|  |  |
| --- | --- |
| **5.** | What is the return type of next() method in the Iterator?Q52 |
| |  | | --- | | A.  double |  |  | | --- | | B.  int |  |  | | --- | | C.  boolean |  |  | | --- | | D.  String |  |  | | --- | | E.  Object | | |
| **6.** | Which method is not available in the Iterator?Q53 |
| |  | | --- | | A.  hasNext() |  |  | | --- | | B.  next() |  |  | | --- | | C.  add() |  |  | | --- | | D.  forEachRemaining | | |

|  |  |
| --- | --- |
| **7.** | Is it possible to read elements in the reverse order by using Iterator?Q54 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **8.** | Is it possible to read elements multiple times by using Iterator?Q55 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |  |
| --- | --- | --- |
| **9.** | Iterator is a class.Q56 | |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | | |
| **10.** | | Iterator is a pointer not a container.Q57 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | | |

|  |  |
| --- | --- |
| **11.** | Which exception occurs while adding an element into ArrayList after getting Iterator and before iterating.Q58 |
| |  | | --- | | A.  ClassCastException |  |  | | --- | | B.  NullPointerException |  |  | | --- | | C.  ConcurrentModificationException |  |  | | --- | | D.  ArithmaticException | | |
| **12.** | Fail Fast Iterators are not allowing concurrent operations.Q59 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **13.** | Fail Safe Iterators are not allowing concurrent operations.Q60 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **14.** | ListIterator extending IteratorQ61 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **15.** | What is the return type of hasPrevious() method in the Iterator?Q62 |
| |  | | --- | | A.  double |  |  | | --- | | B.  int |  |  | | --- | | C.  boolean |  |  | | --- | | D.  String |  |  | | --- | | E.  Object | | |
| **16.** | What is the return type of previous() method in the Iterator?Q63 |
| |  | | --- | | A.  double |  |  | | --- | | B.  int |  |  | | --- | | C.  boolean |  |  | | --- | | D.  String |  |  | | --- | | E.  Object | | |

|  |  |
| --- | --- |
| **17.** | Which method is not available in the Iterator?Q64 |
| |  | | --- | | A.  hasNext() |  |  | | --- | | B.  next() |  |  | | --- | | C.  add() |  |  | | --- | | D.  forEachRemaining |  |  | | --- | | E.  size() | | |
| **18.** | Is it possible to read elements in the reverse order by using ListIterator?Q65 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **19.** | Is it possible to read elements multiple times by using ListIterator?Q66 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **20.** | ListIterator is an interface.Q67 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **21.** | ListIterator is a container not a pointer.Q68 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **22.** | Which exception occurs while adding an element into ArrayList after getting ListIterator and before iterating.Q69 |
| |  | | --- | | A.  ClassCastException |  |  | | --- | | B.  NullPointerException |  |  | | --- | | C.  ConcurrentModificationException |  |  | | --- | | D.  ArithmaticException | | |

|  |  |
| --- | --- |
| **23.** | which method is in only ListIterator but not in IteratorQ70 |
| |  | | --- | | A.  remove() |  |  | | --- | | B.  hasNext() |  |  | | --- | | C.  next() |  |  | | --- | | D.  set() | | |
| **24.** | Enumeration is only for ?Q71 |
| |  | | --- | | A.  reading |  |  | | --- | | B.  reading and remove |  |  | | --- | | C.  reading, adding and remove | | |

|  |  |
| --- | --- |
| **25.** | Entry is a combination of key and value in a MapQ90 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **26.** | What is the return type of entrySet() in HashMapQ91 |
| |  | | --- | | A.  Entry |  |  | | --- | | B.  Set |  |  | | --- | | C.  Key |  |  | | --- | | D.  Value | | |

|  |  |
| --- | --- |
| **27.** | Which method is not from Entry?Q92 |
| |  | | --- | | A.  getKey() |  |  | | --- | | B.  getValue() |  |  | | --- | | C.  getEntryKey() | | |
| **28.** | there is a compare(Object o1, Object o2) method in Integer wrapper classQ93 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **29.** | what is the default sort in TreeMapQ94 |
| |  | | --- | | A.  based on values |  |  | | --- | | B.  based on keys | | |
| **30.** | Which is the method from Comparator?Q95 |
| |  | | --- | | A.  compareTo(Object o1) |  |  | | --- | | B.  compare(Object o1, Object o2) | | |

|  |  |
| --- | --- |
| **31.** | which class is used to sort ArrayList contentQ96 |
| |  | | --- | | A.  TreeList |  |  | | --- | | B.  Collections | | |
| **32.** | How to supply Comparator to the TreeSetQ97 |
| |  | | --- | | A.  to the constructor |  |  | | --- | | B.  to the addAll method | | |

|  |  |
| --- | --- |
| **33.** | which one is proper while implementing compare method to achive sorting based on values?Q98 |
| |  | | --- | | A.  ((Integer)o1.getValue()).compareTo((Integer)o2.getValue()) |  |  | | --- | | B.  ((Integer)o1.getValue()).compare((Integer)o2.getValue()) | | |
| **34.** | Which type is not a Collection typeQ1289 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Set |  |  | | --- | | C.  Queue |  |  | | --- | | D.  Map | | |

|  |  |
| --- | --- |
| **35.** | List is a type of CollectionQ1290 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **36.** | Set is a type of CollectionQ1291 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **37.** | Queue is a type of CollectionQ1292 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **38.** | Map is a type of CollectionQ1293 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **39.** | Which is maintaining oreader of addition. Q1294 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Set |  |  | | --- | | C.  Queue |  |  | | --- | | D.  Map | | |
| **40.** | Which is maintaining unique elementsQ1295 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Set |  |  | | --- | | C.  Queue |  |  | | --- | | D.  Map | | |

|  |  |
| --- | --- |
| **41.** | where elements are adding with a keyQ1296 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Set |  |  | | --- | | C.  Queue |  |  | | --- | | D.  Map | | |
| **42.** | Which can be best to maintain a stackQ1297 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Set |  |  | | --- | | C.  Queue |  |  | | --- | | D.  Map | | |

|  |  |
| --- | --- |
| **43.** | ArrayList comes under?Q1298 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Set |  |  | | --- | | C.  Queue |  |  | | --- | | D.  Map | | |
| **44.** | LinedList mainly comes under?Q1299 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Set |  |  | | --- | | C.  Queue |  |  | | --- | | D.  Map | | |

|  |  |
| --- | --- |
| **45.** | Vector mainly comes under?Q1300 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Set |  |  | | --- | | C.  Queue |  |  | | --- | | D.  Map | | |
| **46.** | Which is alo a QueueQ1301 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  LinkedList |  |  | | --- | | C.  Vector | | |

|  |  |
| --- | --- |
| **47.** | Which is synchronized one?Q1302 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  LinkedList |  |  | | --- | | C.  Vector | | |
| **48.** | Which is the best container while reading elements frequently.?Q1303 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  LinkedList | | |

|  |  |
| --- | --- |
| **49.** | Which is the best container while inserting elements frequently.?Q1304 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  LinkedList | | |
| **50.** | Collections class is used to sort elements of List type.Q1305 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **51.** | Every element is storing with an index inside List type containers.Q1306 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **52.** | every element getting an index automatically under List typeQ1307 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **53.** | Every element is storing with a key inside List type containers.Q1308 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **54.** | Every element is storing with an object association inside List type containers.Q1309 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **55.** | while reading an element from a list type container, we should supply an int value to the get method.Q1310 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **56.** | while reading an element from a list type container, we should supply string value to the get method.Q1311 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **57.** | which class is also queue type from the list type.Q1312 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  LinkedList | | |
| **58.** | This method retrieves, but does not remove, the head (first element) of this queue.Q1313 |
| |  | | --- | | A.  peek() |  |  | | --- | | B.  poll() |  |  | | --- | | C.  get() | | |

|  |  |
| --- | --- |
| **59.** | This method retrieves, and also remove, the head (first element) of this queue.Q1314 |
| |  | | --- | | A.  peek() |  |  | | --- | | B.  poll() |  |  | | --- | | C.  get() | | |
| **60.** | PriorityQueue is a default sorted one.Q1315 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **61.** | PriorityQueue is a default sorted one.Q1316 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **62.** | get method is not available in the PriorityQueueQ1317 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **63.** | Inside a PriorityQueue all the elements will be in the sorted manner.Q1318 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **64.** | Inside a PriorityQueue only head elements is guarenteed in the sorted manner.Q1319 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **65.** | which one not allowing duplicates?Q1320 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Set |  |  | | --- | | C.  Queue |  |  | | --- | | D.  Map | | |
| **66.** | which are the Set type classesQ1321 |
| |  | | --- | | A.  HashSet |  |  | | --- | | B.  LinkedHashSet |  |  | | --- | | C.  ArrayList |  |  | | --- | | D.  HashMap | | |

|  |  |
| --- | --- |
| **67.** | which methods of element class executing to identikfy duplicates?Q1322 |
| |  | | --- | | A.  toString() |  |  | | --- | | B.  hashCode() |  |  | | --- | | C.  equals() |  |  | | --- | | D.  compareTo() | | |
| **68.** | which method is executing first for avoiding duplicates?Q1323 |
| |  | | --- | | A.  toString() |  |  | | --- | | B.  hashCode() |  |  | | --- | | C.  equals() |  |  | | --- | | D.  compareTo() | | |

|  |  |
| --- | --- |
| **69.** | which method is giving perfect results incase of ineqality and may not be perfect incase of equalityQ1324 |
| |  | | --- | | A.  toString() |  |  | | --- | | B.  hashCode() |  |  | | --- | | C.  equals() |  |  | | --- | | D.  compareTo() | | |
| **70.** | which method is giving perfect results incase of ineqality and also equalityQ1325 |
| |  | | --- | | A.  toString() |  |  | | --- | | B.  hashCode() |  |  | | --- | | C.  equals() |  |  | | --- | | D.  compareTo() | | |

|  |  |
| --- | --- |
| **71.** | if two elements hasCode value same, then which method should execute to identify whether both are same or differentQ1326 |
| |  | | --- | | A.  toString() |  |  | | --- | | B.  hashCode() |  |  | | --- | | C.  equals() |  |  | | --- | | D.  compareTo() | | |
| **72.** | hash bucket containing elements which are having same hashCode value.Q1327 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **73.** | Which container used to store elements under one hash bucketQ1328 |
| |  | | --- | | A.  HashSet |  |  | | --- | | B.  LinkedHashSet |  |  | | --- | | C.  ArrayList |  |  | | --- | | D.  HashMap |  |  | | --- | | E.  LinkedList | | |
| **74.** | elements under one hash bucket are storing in the order wiseQ1329 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **75.** | elements under one hash bucket are storing in the randum orderQ1330 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **76.** | equals() method is calling on elements of same bucket one by one in the order wiseQ1331 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **77.** | equals() method is calling on elements of same bucket one by one in the randum orderQ1332 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **78.** | which class is maintaining order of addition under SetQ1333 |
| |  | | --- | | A.  HashSet |  |  | | --- | | B.  LinkedHashSet |  |  | | --- | | C.  ArrayList |  |  | | --- | | D.  HashMap |  |  | | --- | | E.  LinkedList | | |

|  |  |
| --- | --- |
| **79.** | in which containers, element is adding with a key?Q1334 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Set |  |  | | --- | | C.  Queue |  |  | | --- | | D.  Map | | |
| **80.** | Under Map, how many arguments required to put methodQ1335 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  2 |  |  | | --- | | C.  3 |  |  | | --- | | D.  4 | | |

|  |  |
| --- | --- |
| **81.** | what is first argument for put method under MapQ1336 |
| |  | | --- | | A.  key |  |  | | --- | | B.  value or element | | |
| **82.** | what is 2nd argument for put method under MapQ1337 |
| |  | | --- | | A.  key |  |  | | --- | | B.  value or element | | |

|  |  |
| --- | --- |
| **83.** | what is the argument for get method under MapQ1338 |
| |  | | --- | | A.  key |  |  | | --- | | B.  value or element | | |
| **84.** | what is returning by get method under MapQ1339 |
| |  | | --- | | A.  key |  |  | | --- | | B.  value or element | | |

|  |  |
| --- | --- |
| **85.** | What returns get method under map while supplying a key which is not associated to any elementQ1340 |
| |  | | --- | | A.  0 |  |  | | --- | | B.  0.0 |  |  | | --- | | C.  null |  |  | | --- | | D.  false | | |
| **86.** | key can be any type under Map.Q1341 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **87.** | value/element can be any type under Map.Q1342 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **88.** | key always String type under Map.Q1343 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **89.** | how to read all keys of Map?Q1344 |
| |  | | --- | | A.  getKeys() |  |  | | --- | | B.  keySet() |  |  | | --- | | C.  readKeys() | | |
| **90.** | how to read all Entry objects of Map?Q1345 |
| |  | | --- | | A.  entrySet() |  |  | | --- | | B.  getEntries() |  |  | | --- | | C.  readEntries() | | |

|  |  |
| --- | --- |
| **91.** | how to read key from the Entry object under Map?Q1346 |
| |  | | --- | | A.  readKey() |  |  | | --- | | B.  getKey() |  |  | | --- | | C.  getKey() | | |
| **92.** | how to read value from the Entry object under Map?Q1347 |
| |  | | --- | | A.  readValue() |  |  | | --- | | B.  getValue() |  |  | | --- | | C.  get() | | |

|  |  |
| --- | --- |
| **93.** | which one is synchronized under MapQ1348 |
| |  | | --- | | A.  HashMap |  |  | | --- | | B.  Hashtable |  |  | | --- | | C.  LinkedHashMap | | |
| **94.** | which one is maintaining order of addition under MapQ1349 |
| |  | | --- | | A.  HashMap |  |  | | --- | | B.  Hashtable |  |  | | --- | | C.  LinkedHashMap | | |

|  |  |
| --- | --- |
| **95.** | which one allows null as a keyQ1350 |
| |  | | --- | | A.  HashMap |  |  | | --- | | B.  Hashtable | | |
| **96.** | which one allows null as a value/elementQ1351 |
| |  | | --- | | A.  HashMap |  |  | | --- | | B.  Hashtable | | |

|  |  |
| --- | --- |
| **97.** | Unser Map duplicate keys are allowedQ1352 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **98.** | Unser Map duplicate values/elements are allowedQ1353 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |  |
| --- | --- | --- |
| **99.** | Iterator is an interface.Q1354 | |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | | |
| **100.** | | Iterator is a container and not a reference to ArrayList objectQ1355 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | | |

|  |  |
| --- | --- |
| **101.** | we can iterate elements any number of times by using one IteratorQ1356 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **102.** | we can't iterate elements in the reverse order by using IteratorQ1357 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **103.** | we can remove iterating element while iterating through Iterator. This remove will remove element from an ArrayList itself.Q1358 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **104.** | ListIterator extending IteratorQ1359 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **105.** | ListIterator is not a container and just a reference to ArrayList objectQ1360 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **106.** | we can iterate elements any number of times by using one IteratorQ1361 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **107.** | we can't iterate elements in the reverse order by using ListIteratorQ1362 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **108.** | we can remove iterating element while iterating through ListIterator. This remove will remove element from an ArrayList itself.Q1363 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **109.** | we can add element into ArrayList through ListIteratorQ1364 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **110.** | we can add element into ArrayList through IteratorQ1365 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **111.** | we can replace element from an ArrayList through IteratorQ1366 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **112.** | we can replace element from an ArrayList through ListIteratorQ1367 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **113.** | we can find next element index through ListIteratorQ1368 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **114.** | we can find next element index through IteratorQ1369 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **115.** | by default Iterator is a fail safe oneQ1370 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **116.** | by default Iterator is a fail fast oneQ1371 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **117.** | by default ListIterator is a fail safe oneQ1372 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **118.** | by default ListIterator is a fail fast oneQ1373 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **119.** | is it possible to add any element into ArrayList after getting Iterator and before iterating.Q1374 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **120.** | is it possible to add any element into ArrayList after getting ListIterator and before iterating. (using add method of ArrayList)Q1375 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **121.** | is it possible to add any element into ArrayList after getting ListIterator and before iterating. (using add method of ListIterator)Q1376 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **122.** | Which exception occures in case of fail fast iterators. Q1377 |
| |  | | --- | | A.  NullPinterException |  |  | | --- | | B.  CuncurrentModificationException |  |  | | --- | | C.  NumberFormatException | | |

|  |  |
| --- | --- |
| **123.** | Which class is used to sort elements from List type of containers. Q1378 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  LinkedList |  |  | | --- | | C.  Vector |  |  | | --- | | D.  Collection |  |  | | --- | | E.  Collections | | |
| **124.** | Which class is used to sort elements from Set type of containers. Q1379 |
| |  | | --- | | A.  HashSet |  |  | | --- | | B.  LinkedHashSet |  |  | | --- | | C.  TreeSet |  |  | | --- | | D.  Collection |  |  | | --- | | E.  Collections | | |

|  |  |
| --- | --- |
| **125.** | Which class is used to sort entries from Map type of containers. Q1380 |
| |  | | --- | | A.  HashMap |  |  | | --- | | B.  LinkedHashMap |  |  | | --- | | C.  TreeMap |  |  | | --- | | D.  Hashtable |  |  | | --- | | E.  Collections | | |
| **126.** | Which class is used to sort elements from Queue type of containers. Q1381 |
| |  | | --- | | A.  LinkedList |  |  | | --- | | B.  PriorityQueue | | |

|  |  |
| --- | --- |
| **127.** | Is it required all elements should be same type in a container while sorting all elements from a container?Q1382 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **128.** | Is it required all elements should be not null in a container while sorting all elements from a container?Q1383 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **129.** | Is it required all elements should be type of Comparable in a container while sorting all elements from a container and there is no separate Comparator supplying?Q1384 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **130.** | Is it required all elements should be type of Comparable in a container while sorting all elements from a container and there is a separate Comparator supplying?Q1385 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **131.** | Comparable is an interfaceQ1386 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **132.** | Comparator is an interfaceQ1387 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **133.** | Which method available in Comparable?Q1388 |
| |  | | --- | | A.  compareTo(Object obj) |  |  | | --- | | B.  compare(Object obj1, Object obj2) | | |
| **134.** | Which method available in Comparator?Q1389 |
| |  | | --- | | A.  compareTo(Object obj) |  |  | | --- | | B.  compare(Object obj1, Object obj2) | | |

|  |  |
| --- | --- |
| **135.** | Comparable is a marker interfaceQ1390 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **136.** | Comparable is a functional interfaceQ1391 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **137.** | Comparator is a marker interfaceQ1392 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **138.** | Comparator is a functional interfaceQ1393 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **139.** | If class is with a single field, then wich interface is most suitable to achieve sorting?Q1394 |
| |  | | --- | | A.  Comparable |  |  | | --- | | B.  Comparator | | |
| **140.** | If class is with multiple fields, then wich interface is most suitable to achieve sorting?Q1395 |
| |  | | --- | | A.  Comparable |  |  | | --- | | B.  Comparator | | |

|  |  |
| --- | --- |
| **141.** | wich interface implemented by wrapper classes.Q1396 |
| |  | | --- | | A.  Comparable |  |  | | --- | | B.  Comparator | | |
| **142.** | wich interface implemented by String class.Q1397 |
| |  | | --- | | A.  Comparable |  |  | | --- | | B.  Comparator | | |

|  |  |
| --- | --- |
| **143.** | How to supply Comparator type to Collections's sort method?Q1398 |
| |  | | --- | | A.  as first arg |  |  | | --- | | B.  as a 2nd arg | | |
| **144.** | Is it possible to sort elements of a container which contains null value?Q1399 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |

|  |  |
| --- | --- |
| **145.** | Is it possible to sort elements of a container which contains different type of elements?Q1400 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **146.** | How to supply Comparator type to PriorityQueue?Q1401 |
| |  | | --- | | A.  as an arg to add method |  |  | | --- | | B.  as an arg to constructor | | |

|  |  |
| --- | --- |
| **147.** | Which class is used to sort elements from Set type of containers. Q1402 |
| |  | | --- | | A.  HashSet |  |  | | --- | | B.  LinkedHashSet |  |  | | --- | | C.  TreeSet |  |  | | --- | | D.  Collection |  |  | | --- | | E.  Collections | | |
| **148.** | Which class is used to sort entries from Map type of containers. Q1403 |
| |  | | --- | | A.  HashMap |  |  | | --- | | B.  LinkedHashMap |  |  | | --- | | C.  TreeMap |  |  | | --- | | D.  Hashtable |  |  | | --- | | E.  Collections | | |

|  |  |
| --- | --- |
| **149.** | How to supply Comparator to TreeSet to achive sorting.Q1404 |
| |  | | --- | | A.  an arg to sort method |  |  | | --- | | B.  an arg to constructor | | |
| **150.** | How to supply Comparator to TreeMap to achive sorting.Q1405 |
| |  | | --- | | A.  an arg to sort method |  |  | | --- | | B.  an arg to constructor | | |

|  |  |
| --- | --- |
| **151.** | Based on which quantity sorting is happening in TreeMap?Q1406 |
| |  | | --- | | A.  key |  |  | | --- | | B.  value or element | | |
| **152.** | What is the condition to be followed if TreeMap is not taking Comparator?Q1407 |
| |  | | --- | | A.  key should be a Comparable type |  |  | | --- | | B.  value or element should be a Comparable type | | |

|  |  |
| --- | --- |
| **153.** | how to read all keys of Map?Q1408 |
| |  | | --- | | A.  getKeys() |  |  | | --- | | B.  keySet() |  |  | | --- | | C.  readKeys() | | |
| **154.** | how to read all Entry objects of Map?Q1409 |
| |  | | --- | | A.  entrySet() |  |  | | --- | | B.  getEntries() |  |  | | --- | | C.  readEntries() | | |

|  |  |
| --- | --- |
| **155.** | how to read key from the Entry object under Map?Q1410 |
| |  | | --- | | A.  readKey() |  |  | | --- | | B.  getKey() |  |  | | --- | | C.  getKey() | | |
| **156.** | how to read value from the Entry object under Map?Q1411 |
| |  | | --- | | A.  readValue() |  |  | | --- | | B.  getValue() |  |  | | --- | | C.  get() | | |

|  |  |
| --- | --- |
| **157.** | in which class, synchronizedSet() available?Q1412 |
| |  | | --- | | A.  SortedSet |  |  | | --- | | B.  TreeSet |  |  | | --- | | C.  LinkedHashSet |  |  | | --- | | D.  Collections | | |
| **158.** | Which class is used to make List as a synchronized one.Q1413 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  TreeSet |  |  | | --- | | C.  LinkedList |  |  | | --- | | D.  Collections | | |

|  |  |
| --- | --- |
| **159.** | How to make HashMap as a synhronized one?Q1414 |
| |  | | --- | | A.  HashMap.synchronizedMap(); |  |  | | --- | | B.  Collections.synchronizedMap(Map map); | | |
| **160.** | class M1  {  public static void main(String[] args)  {  String s1 = "hello";  System.out.println(s1);  String s2 = new String("hello");  System.out.println(s2);  }  }  Q1554 |
| |  | | --- | | A.  hello  hello |  |  | | --- | | B.  hello |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  none | | |

|  |  |
| --- | --- |
| **161.** | class M2  {  public static void main(String[] args)  {  String s1 = "hello";  String s2 = "hello";  System.out.println(s1 == s2);  }  }  Q1555 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false |  |  | | --- | | C.  Compilation error | | |
| **162.** | class M3  {  public static void main(String[] args)  {  String s1 = new String("hello");  String s2 = new String("hello");  System.out.println(s1 == s2);  }  }  Q1556 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **163.** | class M4  {  public static void main(String[] args)  {  String s1 = "hello";  String s2 = "Hello";  String s3 = new String("hello");  String s4 = new String("hello");  System.out.println(s3 .equals(s4));  }  }  Q1557 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false |  |  | | --- | | C.  Compilation error | | |
| **164.** | class M5  {  public static void main(String[] args)  {  String s1 = "hello";  int i = s1.length();  System.out.println(i);  System.out.println(s1.length());  }  }  Q1558 |
| |  | | --- | | A.  5  5 |  |  | | --- | | B.  4  4 |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **165.** | class M6  {  public static void main(String[] args)  {  String s1 = "a c";  String s2 = " a c ";  System.out.println(s1.length());  System.out.println(s2.length());  }  }  Q1559 |
| |  | | --- | | A.  5  3 |  |  | | --- | | B.  3  5 |  |  | | --- | | C.  Compilation error | | |
| **166.** | class M7  {  public static void main(String[] args)  {  String s1 = "Hi India";  String s2 = "Hi "India";  System.out.println(s1);  System.out.println(s2);  }  }  Q1560 |
| |  | | --- | | A.  Hi India  Hi "India |  |  | | --- | | B.  Hi India  Hi India |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **167.** | class M8  {  public static void main(String[] args)  {  String s1 = "Hi India";  String s2 = "Hi \"India\"";  System.out.println(s1.length());  System.out.println(s2.length());  }  }  Q1561 |
| |  | | --- | | A.  10  8 |  |  | | --- | | B.  8  10 |  |  | | --- | | C.  Compilation error | | |
| **168.** | class M9  {  public static void main(String[] args)  {  String s1 = "hello n hello";  String s2 = "hello \n hello";  System.out.println(s1);  System.out.println(s2);  }  }  Q1562 |
| |  | | --- | | A.  hello n hello  hello  hello |  |  | | --- | | B.  hello n hello  hello |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **169.** | class M10  {  public static void main(String[] args)  {  String s1 = "abc \ xyz";  System.out.println(s1);  }  }  // Will it compiles fine or not?Q1563 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |
| **170.** | class M11  {  public static void main(String[] args)  {  String s1 = "hello \\ xyz";  System.out.println(s1);  }  }  // will it com[piles fine or not?Q1564 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |

|  |  |
| --- | --- |
| **171.** | class M12  {  public static void main(String[] args)  {  String path = "D:\JDK8.0\bin";  System.out.println(path);  }  }  Q1565 |
| |  | | --- | | A.  D:\JDK8.0\bin |  |  | | --- | | B.  runtime error |  |  | | --- | | C.  Compilation error | | |
| **172.** | class M13  {  public static void main(String[] args)  {  String path = "D:\\JDK8.0\\bin";  System.out.println(path);  }  }  Q1566 |
| |  | | --- | | A.  D:\JDK8.0\bin |  |  | | --- | | B.  runtime error |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **173.** | class M14  {  public static void main(String[] args)  {  String path = "D:/JDK8.0/bin";  System.out.println(path);  }  }  Q1567 |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  runtime error |  |  | | --- | | C.  D:/JDK8.0/bin | | |
| **174.** | class M15  {  public static void main(String[] args)  {  String s1 = "abc";  s1 = s1 + "xyz";  System.out.println(s1);  }  }  Q1568 |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  runtime error |  |  | | --- | | C.  abcxyz |  |  | | --- | | D.  abc + xyz |  |  | | --- | | E.  None | | |

|  |  |
| --- | --- |
| **175.** | class M16  {  public static void main(String[] args)  {  String s1 = "abc";  String s2 = null;  String s3 = s1 + s2;  String s4 = s2 + s1;  System.out.println(s1);  System.out.println(s2);  System.out.println(s3);  System.out.println(s4);  }  }  Q1569 |
| |  | | --- | | A.  abc  null  abcnull  nullabc |  |  | | --- | | B.  abc  null  abcnull  abcnull |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  null  abc  abcnull  nullabc | | |
| **176.** | class M17  {  public static void main(String[] args)  {  String s1 = null;  String s2 = s1 + null;  String s3 = null + s1;  System.out.println(s1);  System.out.println(s2);  System.out.println(s3);  }  }  Q1570 |
| |  | | --- | | A.  null  null  nullnull |  |  | | --- | | B.  null  nullnull  nullnull |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **177.** | class M18  {  public static void main(String[] args)  {  String s1 = null + null;  System.out.println(s1);  }  }  Q1571 |
| |  | | --- | | A.  nullnull |  |  | | --- | | B.  null + null |  |  | | --- | | C.  Compilation error | | |
| **178.** | class M19  {  public static void main(String[] args)  {  String s1 = "abc" + 5 + 6;  String s2 = 5 + "abc" + 6;  String s3 = 5 + 6 + "abc";  System.out.println(s1);  System.out.println(s2);  System.out.println(s3);  }  }  Q1572 |
| |  | | --- | | A.  abc56  5abc6  11abc |  |  | | --- | | B.  abc11  5abc6  11abc |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **179.** | class M20  {  public static void main(String[] args)  {  String s1 = null + null;  String s2 = 5 + 9;  String s3 = null + "";  String s4 = "" + null;  System.out.println(s1);  System.out.println(s2);  System.out.println(s3);  System.out.println(s4);  }  }  Q1573 |
| |  | | --- | | A.  nullnull  14  null  null |  |  | | --- | | B.  Compilation error |  |  | | --- | | C.  runtime error | | |
| **180.** | class M21  {  public static void main(String[] args)  {  String s1 = null;  System.out.println(s1.length());  }  }  // will it compiles fine or not?Q1574 |
| |  | | --- | | A.  yes | | |

|  |  |
| --- | --- |
| **181.** | class M22  {  public static void main(String[] args)  {  String s1 = NULL;  System.out.println("Hello World!"s1.length();  }  }  //will it compiles fine or not?Q1575 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  No | | |
| **182.** | .class M23  {  public static void main(String[] args)  {  String s1 = "null";  System.out.println(s1.length());  }  }  Q1576 |
| |  | | --- | | A.  4 |  |  | | --- | | B.  NullPointerException |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **183.** | class M24  {  public static void main(String[] args)  {  String s1 = "xyz";  System.out.println(s1);  s1.concat("hello");  System.out.println(s1);  }  }  Q1577 |
| |  | | --- | | A.  xyz  xyz |  |  | | --- | | B.  xyzxyz |  |  | | --- | | C.  Compilation error | | |
| **184.** | class M25  {  public static void main(String[] args)  {  String s1 = "xyz";  System.out.println(s1);  String s2 = s1.concat("hello");  System.out.println(s1);  System.out.println(s2);  }  }  Q1578 |
| |  | | --- | | A.  xyzxyz  xyzhello |  |  | | --- | | B.  xyz  xyz  xyz  hello |  |  | | --- | | C.  xyz  xyz  xyzhello |  |  | | --- | | D.  Compilation error | | |

|  |  |
| --- | --- |
| **185.** | package StringBuffer;  public class S1 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer();  sb.append("hello");  sb.append("test");  sb.append("123");  System.out.println(sb);  }  }  Q1634 |
| |  | | --- | | A.  true  true |  |  | | --- | | B.  false  false |  |  | | --- | | C.  true  false |  |  | | --- | | D.  false  true | | |
| **186.** | package StringBuffer;  public class S13 {  public static void main(String[] args) {  StringBuilder sb = new StringBuilder("hellotoall");  System.out.println(sb);  }  }  Q1635 |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  123 |  |  | | --- | | C.  hellotest123 | | |

|  |  |
| --- | --- |
| **187.** | package StringBuffer;  public class S3 {  public static void main(String[] args) {  StringBuffer sb1 = new StringBuffer();  sb1.append("hello");    System.out.println(sb1.length());  System.out.println(sb1.capacity());  }  }  Q1636 |
| |  | | --- | | A.  5  16 |  |  | | --- | | B.  5  34 |  |  | | --- | | C.  5  32 | | |
| **188.** | package StringBuffer;  public class S4 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer();  System.out.println(sb.capacity());  System.out.println(sb.length());  System.out.println("---------------------");  sb.append("123456789987654321");  System.out.println(sb.length());  System.out.println(sb.capacity());  }  }  Q1637 |
| |  | | --- | | A.  16  1  ---------------------  18  34 |  |  | | --- | | B.  16  0  ---------------------  18  32 |  |  | | --- | | C.  18  34  ---------------------  16  0 |  |  | | --- | | C.  16  0  ---------------------  18  34 | | |

|  |  |
| --- | --- |
| **189.** | package StringBuffer;  public class S5 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer(2000);  System.out.println(sb.capacity());  System.out.println(sb.length());  System.out.println("---------------------");  sb.append("123456789987654321");  System.out.println(sb.length());  System.out.println(sb.capacity());  }  }  Q1638 |
| |  | | --- | | A.  2000  0  ---------------------  18  2000 |  |  | | --- | | B.  2000  1  ---------------------  18  2000 |  |  | | --- | | C.  Compilation error | | |
| **190.** | package StringBuffer;  public class S6 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("test");  System.out.println(sb.capacity());  System.out.println(sb.length());  System.out.println("---------------------");  sb.append("123456789987654321");  System.out.println(sb.length());  System.out.println(sb.capacity());  System.out.println("---------------------");  sb.trimToSize();  System.out.println(sb.length());  System.out.println(sb.capacity());  }  }  Q1639 |
| |  | | --- | | A.  16  4  ---------------------  22  44  ---------------------  22  22 |  |  | | --- | | B.  20  4  ---------------------  22  42  ---------------------  22  22 |  |  | | --- | | C.  20  4  ---------------------  22  42  ---------------------  22  44 | | |

|  |  |
| --- | --- |
| **191.** | package StringBuffer;  public class S7 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("test");  System.out.println(sb.capacity());  System.out.println(sb.length());  System.out.println("---------------------");  sb.append("123456789987654321");  System.out.println(sb.length());  System.out.println(sb.capacity());  System.out.println("---------------------");  sb.trimToSize();  System.out.println(sb.length());  System.out.println(sb.capacity());  }  }  Q1640 |
| |  | | --- | | A.  20  4  ---------------------  22  42  ---------------------  22  22 |  |  | | --- | | B.  16  4  ---------------------  22  44  ---------------------  22  22 |  |  | | --- | | C.  20  4  ---------------------  22  42  ---------------------  22  44 | | |
| **192.** | package StringBuffer;  public class S8 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("test");  System.out.println(sb);  sb.reverse();  System.out.println(sb);  }  }  Q1641 |
| |  | | --- | | A.  tset |  |  | | --- | | B.  test  tset |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **193.** | package StringBuffer;  public class S9 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("test");  System.out.println(sb);  sb.deleteCharAt(2);  System.out.println(sb);  }  }  Q1642 |
| |  | | --- | | A.  test  tet |  |  | | --- | | B.  tet |  |  | | --- | | C.  Compilation error | | |
| **194.** | package StringBuffer;  public class S10 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("hellotoall");  System.out.println(sb);  sb.delete(2, 6);  System.out.println(sb);  }  }  Q1643 |
| |  | | --- | | A.  hellotoall  heloall |  |  | | --- | | B.  hellotoall  hetoall |  |  | | --- | | C.  hellotoall  heoall |  |  | | --- | | D.  Compilation error | | |

|  |  |
| --- | --- |
| **195.** | package StringBuffer;  public class S11 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("hellotoall");  System.out.println(sb);  sb.delete(2, 30);  System.out.println(sb);  }  }  Q1644 |
| |  | | --- | | A.  hellotoall  hel |  |  | | --- | | B.  StringIndexOutOfBoundsException |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  hellotoall  he | | |
| **196.** | package StringBuffer;  public class S12 {  public static void main(String[] args) {  StringBuffer sb = new StringBuffer("hellotoall");  System.out.println(sb);  sb.delete(20, 30);  System.out.println(sb);  }  }  Q1645 |
| |  | | --- | | A.  hellotoall  StringIndexOutOfBoundsException |  |  | | --- | | B.  StringIndexOutOfBoundsException |  |  | | --- | | C.  hellotoall | | |

|  |  |
| --- | --- |
| **197.** | package StringBuffer;  public class S13 {  public static void main(String[] args) {  StringBuilder sb = new StringBuilder("hellotoall");  System.out.println(sb);  }  }  Q1646 |
| |  | | --- | | A.  hellotoall |  |  | | --- | | B.  No output |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  StringOutOfBoundsException | | |
| **198.** | package formatter;  public class S1 {  public static void main(String[] args) {  String s1 = "hello";  String s2 = String.format("==> %s", s1);  System.out.println(s1);  System.out.println(s2);  }  }  Q1647 |
| |  | | --- | | A.  hello==> hello <== |  |  | | --- | | B.  hello  ==> hello |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **199.** | package formatter;  public class S2 {  public static void main(String[] args) {  String s1 = "hello";  String s2 = String.format("==> %s <==", s1);  System.out.println(s1);  System.out.println(s2);  }  }  Q1648 |
| |  | | --- | | A.  hello  ==> hello <== |  |  | | --- | | B.  hello==> hello <== |  |  | | --- | | C.  Compilation error | | |
| **200.** | package formatter;  public class S3 {  public static void main(String[] args) {  String s1 = "hello";  String s2 = String.format(" (%s) and (%d) ans (%f)", s1, 100, 4.5);  System.out.println(s1);  System.out.println(s2);  }  }  Q1649 |
| |  | | --- | | A.  hello  (hello) and (100) ans (4.500000) |  |  | | --- | | B.  hello (hello) and (100) ans (4.500000) |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **201.** | package formatter;  public class S4 {  public static void main(String[] args) {  String s1 = "hello";  int i = 100;  double d = 2.5;  char ch = 'k';  boolean b = true;  System.out.printf("i value is %d ", i);  System.out.printf("j value is %f ", d);  System.out.printf("b value is %b ", b);  System.out.printf("s1 value is %s ", s1);  System.out.printf("ch value is %c ", ch);  }  }  Q1650 |
| |  | | --- | | A.  i value is 100  j value is 2.5  b value is true  s1 value is hello  ch value is k |  |  | | --- | | B.  i value is 100j value is 2.5b value is trues1 value is helloch value is k |  |  | | --- | | C.  Compilation error | | |
| **202.** | package formatter;  public class S5 {  public static void main(String[] args) {  String s1 = "hello";  int i = 100;  double d = 2.5;  char ch = 'k';  boolean b = true;  System.out.printf("i value is %d ", i);  System.out.printf("\nj value is %f ", d);  System.out.printf("\nb value is %b ", b);  System.out.printf("\ns1 value is %s ", s1);  System.out.printf("\nch value is %c ", ch);  }  }  Q1651 |
| |  | | --- | | A.  i value is 100  j value is 2.5  b value is true  s1 value is hello  ch value is k |  |  | | --- | | B.  i value is 100j value is 2.5b value is trues1 value is helloch value is k |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **203.** | package formatter;  public class S6 {  public static void main(String[] args) {  String s1 = "hello";  int i = 100;  double d = 2.5;  char ch = 'k';  boolean b = true;  System.out.printf("i value is %s ", i);  System.out.printf("\nj value is %s ", d);  System.out.printf("\nb value is %s ", b);  System.out.printf("\ns1 value is %s ", s1);  System.out.printf("\nch value is %s ", ch);  }  }  Q1652 |
| |  | | --- | | A.  i value is 100  j value is 2.5  b value is true  s1 value is hello  ch value is k |  |  | | --- | | B.  i value is 100j value is 2.5b value is trues1 value is helloch value is k |  |  | | --- | | C.  Compilation error | | |
| **204.** | package formatter;  public class S7 {  public static void main(String[] args) {  double d = 2.5;  System.out.printf("\nj value is %d ", d);    }  }  // Will it compiles fine or notQ1653 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |

|  |  |
| --- | --- |
| **205.** | package formatter;  public class S8 {  public static void main(String[] args) {  int i = 100;  double j = 2.5;  System.out.printf("i value is %d and j value is %f",i, j);    }  }  Q1654 |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  i value is 100 and d value is 2.500000 |  |  | | --- | | C.  i value is 100 and d value is 2.5 | | |
| **206.** | package formatter;  public class S9 {  public static void main(String[] args) {  int i = 100;  double j = 2.5;  System.out.printf("j value is %2$f and j value is %1$d",i, j);    }  }  Q1655 |
| |  | | --- | | A.  j value is 2.500000 and j value is 100 |  |  | | --- | | B.  j value is 2.5 and j value is 100 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **207.** | package formatter;  public class S10 {  public static void main(String[] args) {  int num = 123456;  String s1 = String.format("num is (%d)", num);  System.out.println(s1);  }  }  Q1656 |
| |  | | --- | | A.  123456 |  |  | | --- | | B.  (123456) |  |  | | --- | | C.  num is (123456) |  |  | | --- | | D.  Compilation error | | |
| **208.** | package formatter;  public class S11 {  public static void main(String[] args) {  int num = 123456;  String s1 = String.format("num is (%020d)", num);  System.out.println(s1);  }  }  Q1657 |
| |  | | --- | | A.  num is (0000000000000,123456) |  |  | | --- | | B.  num is (0000000000000123456) |  |  | | --- | | C.  num is (0000000000000123,456) |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **209.** | package formatter;  public class S12 {  public static void main(String[] args) {  int num = 123456;  String s1 = String.format("num is (%-20d)", num);  System.out.println(s1);  }  }  Q1658 |
| |  | | --- | | A.  num is (123456) |  |  | | --- | | B.  num is (123456 ) |  |  | | --- | | C.  Compilation error | | |
| **210.** | package formatter;  public class S13 {  public static void main(String[] args) {  int num = 123456;  String s1 = String.format("num is (%-020d)", num);  System.out.println(s1);  }  }  // Will it compiles fine or not?Q1659 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |

|  |  |
| --- | --- |
| **211.** | package formatter;  public class S14 {  public static void main(String[] args) {  int num = 123456;  String s1 = String.format("num is (%0,20d)", num);  System.out.println(s1);  }  }  Q1660 |
| |  | | --- | | A.  num is (0000000000000,123456) |  |  | | --- | | B.  num is (0000000000000123456) |  |  | | --- | | C.  num is (0000000000000123,456) |  |  | | --- | | C.  Compilation error | | |
| **212.** | package formatter;  public class S15 {  public static void main(String[] args) {  int num = -123456;  String s1 = String.format("num is %(d", num);  System.out.println(s1);  }  }  Q1661 |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  num is (123456) |  |  | | --- | | C.  123456 | | |

|  |  |
| --- | --- |
| **213.** | package formatter;  public class S16 {  public static void main(String[] args) {  double num = 1234.5678;  String s1 = String.format("num is %.3f", num);  System.out.println(s1);  }  }  Q1662 |
| |  | | --- | | A.  num is 1234.568 |  |  | | --- | | B.  num is 1234.567 |  |  | | --- | | C.  num is 12345.678 | | |
| **214.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S1 {  public static void main(String[] args) {  String src = "hello and hello and hello Again and Again";  String regex = "and";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  }  Q1663 |
| |  | | --- | | A.  6:and  16:and  32:and |  |  | | --- | | B.  Compilation error |  |  | | --- | | C.  and  and  and |  |  | | --- | | D.  6:  16:  32: | | |

|  |  |
| --- | --- |
| **215.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S2 {  public static void main(String[] args) {  String src = "hello and hello and hello Again and Again";  String regex = "hello";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  }  Q1664 |
| |  | | --- | | A.  hello  hello  hello |  |  | | --- | | B.  0:hello  10:hello  20:hello |  |  | | --- | | C.  0:  10:  20: |  |  | | --- | | D.  Compilation error | | |
| **216.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S3 {  public static void main(String[] args) {  String src = "hello and hello and hello Again and Again";  String regex = "a";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  }  Q1665 |
| |  | | --- | | A.  6:a  16:a  28:a  32:a  38:a |  |  | | --- | | B.  Compilation error |  |  | | --- | | C.  6:  16:  28:  32:  38: | | |

|  |  |
| --- | --- |
| **217.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S4 {  public static void main(String[] args) {  String src = "hello and hello and hello Again and Again";  String regex = "[ali]";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  }  Q1666 |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  No output |  |  | | --- | | C.  2:l  3:l  6:a  12:l  13:l  16:a  22:l  23:l  28:a  29:i  32:a  38:a  39:i | | |
| **218.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S4 {  public static void main(String[] args) {  String src = "java8, java10, java11, java12, oracle18";  String regex = "\\s";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  }  Q1667 |
| |  | | --- | | A.  6:  14:  15:  16:  17:  18:  26:  27:  28:  29:  30:  31:  32:  33:  41:  42:  43:  44:  45:  46:  47:  48:  49: |  |  | | --- | | B.  6:  14:  26:  41: |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **219.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S4 {  public static void main(String[] args) {  String src = "java8, java10, java11, java12, oracle18";  String regex = "\\s+";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  }  Q1668 |
| |  | | --- | | A.  6:  14:  15:  16:  17:  18:  26:  27:  28:  29:  30:  31:  32:  33:  41:  42:  43:  44:  45:  46:  47:  48:  49: |  |  | | --- | | B.  6:  14:  26:  41: |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | |
| **220.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S5{  public static void main(String[] args) {  String src = "java8, java10, java11, java12, oracle18";  String regex = "\\d";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  }  Q1669 |
| |  | | --- | | A.  4:8  11:10  23:11  38:12  56:18 |  |  | | --- | | B.  Compilation error |  |  | | --- | | C.  4:8  11:1  12:0  23:1  24:1  38:1  39:2  56:1  57:8 | | |

|  |  |
| --- | --- |
| **221.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S5{  public static void main(String[] args) {  String src = "java8, java10, java11, java12, oracle18";  String regex = "\\d+";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  }  Q1670 |
| |  | | --- | | A.  4:8  11:10  23:11  38:12  56:18 |  |  | | --- | | B.  Compilation error |  |  | | --- | | C.  4:8  11:1  12:0  23:1  24:1  38:1  39:2  56:1  57:8 | | |
| **222.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S6 {  public static void main(String[] args) {  String src = "java@java.com";  String regex = "@";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  while(matches.find()) {  System.out.println(matches.start() + ":" + matches.group());  }  }  }  Q1671 |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  4:@ |  |  | | --- | | C.  4: |  |  | | --- | | D.  @ | | |

|  |  |
| --- | --- |
| **223.** | package regularExpression;  import java.util.regex.Matcher;  import java.util.regex.Pattern;  public class S6 {  public static void main(String[] args) {  String src = "java@java.com";  String regex = ".";  //String regex = "@";  Pattern pattern = Pattern.compile(regex);  Matcher matches = pattern.matcher(src);  System.out.println(matches.find());  }  }  Q1672 |
| |  | | --- | | A.  true |  |  | | --- | | B.  false | | |
| **224.** | package regularExpression;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Date date = new Date();  System.out.println(date);  }  }  Q1673 |
| |  | | --- | | A.  Tue Dec 17 14:07:34 IST 2019 |  |  | | --- | | B.  Thu Jan 01 05:30:00 IST 1970 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  No output | | |

|  |  |
| --- | --- |
| **225.** | package regularExpression;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Date date = new Date(0);  System.out.println(date);  }  }  Q1674 |
| |  | | --- | | A.  Tue Dec 17 14:07:34 IST 2019 |  |  | | --- | | B.  Thu Jan 01 05:30:00 IST 1970 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  No output | | |
| **226.** | package regularExpression;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Date date = new Date(1000\*60\*24);  System.out.println(date);  }  }  Q1675 |
| |  | | --- | | A.  Tue Dec 17 14:07:34 IST 2019 |  |  | | --- | | B.  Thu Jan 01 05:30:00 IST 1970 |  |  | | --- | | C.  Fri Jan 02 05:30:00 IST 1970 | | |

|  |  |
| --- | --- |
| **227.** | package regularExpression;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Date date = new Date(-1000\*60\*60\*24);  System.out.println(date);  }  }  Q1676 |
| |  | | --- | | A.  Wed Dec 31 05:30:00 IST 1969 |  |  | | --- | | B.  Thu Jan 01 05:30:00 IST 1970 |  |  | | --- | | C.  Fri Jan 02 05:30:00 IST 1970 | | |
| **228.** | package regularExpression;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  Date date = cal.getTime();  System.out.println(date);  }  }  Q1677 |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  Tue Dec 17 14:14:38 IST 2019 |  |  | | --- | | C.  Fri Jan 02 05:30:00 IST 1970 | | |

|  |  |
| --- | --- |
| **229.** | package regularExpression;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  cal.add(Calendar.DATE, 20);  Date date = cal.getTime();  System.out.println(date);  }  }  Q1678 |
| |  | | --- | | A.  Mon Jan 06 14:18:12 IST 2020 |  |  | | --- | | B.  Fri Dec 20 14:14:38 IST 2019 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | |
| **230.** | package regularExpression;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  cal.add(Calendar.MONTH, -5);  Date date = cal.getTime();  System.out.println(date);  }  }  Q1679 |
| |  | | --- | | A.  Mon Jan 06 14:18:12 IST 2020 |  |  | | --- | | B.  Fri Dec 20 14:14:38 IST 2019 |  |  | | --- | | C.  Wed Jul 17 14:20:28 IST 2019 | | |

|  |  |
| --- | --- |
| **231.** | package regularExpression;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  cal.add(Calendar.YEAR, 10);  Date date = cal.getTime();  System.out.println(date);  }  }  Q1680 |
| |  | | --- | | A.  Mon Dec 17 14:21:55 IST 2029 |  |  | | --- | | B.  Fri Dec 20 14:14:38 IST 2019 |  |  | | --- | | C.  Wed Jul 17 14:20:28 IST 2019 | | |
| **232.** | package regularExpression;  import java.text.DateFormat;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  Date date = cal.getTime();  System.out.println(date);  DateFormat df = DateFormat.getInstance();  String s1 = df.format(date);  System.out.println(s1);  }  }  Q1681 |
| |  | | --- | | A.  Tue Dec 17 14:24:18 IST 2019  12/17/19 2:24 AM |  |  | | --- | | B.  Tue Dec 17 14:24:18 IST 2019  12/17/19 2:24 PM |  |  | | --- | | C.  Tue Dec 17 14:24:18 IST 2020  12/17/19 2:24 PM | | |

|  |  |
| --- | --- |
| **233.** | package regularExpression;  import java.text.DateFormat;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  Date date = cal.getTime();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.SHORT);  String s1 = df.format(date);  System.out.println(s1);  }  }  Q1682 |
| |  | | --- | | A.  Tue Dec 17 14:24:18 IST 2019  12/17/19 2:24 AM |  |  | | --- | | B.  Tue Dec 17 14:27:13 IST 2019  12/17/19 |  |  | | --- | | B.  Tue Dec 17 14:27:13 IST 2019  12/17/20 | | |
| **234.** | package regularExpression;  import java.text.DateFormat;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {    Calendar cal = Calendar.getInstance();  Date date = cal.getTime();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.LONG);  String s1 = df.format(date);  System.out.println(s1);  }  }  Q1683 |
| |  | | --- | | A.  Tue Dec 17 14:28:33 IST 2019  December 17, 2019 |  |  | | --- | | B.  Mon 14:28:33 IST 2020  December 17, 2020 |  |  | | --- | | C.  Tue Dec 17 14:28:33 IST 2018  December 17, 2018 | | |

|  |  |
| --- | --- |
| **235.** | package regularExpression;  import java.text.DateFormat;  import java.text.ParseException;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  Date date = cal.getTime();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.FULL);  String s1 = df.format(date);  System.out.println(s1);    try {  Date date2 = df.parse(s1);  System.out.println(date2);  }  catch(ParseException ex) {  ex.printStackTrace();  }  }  }  Q1684 |
| |  | | --- | | A.  Tue Dec 17 14:32:08 IST 2019  Tuesday, December 17, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 14:32:08 IST 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Compilation error | | |
| **236.** | package regularExpression;  import java.text.DateFormat;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Date date = new Date();  System.out.println(date);  SimpleDateFormat sdf = new SimpleDateFormat("dd----MM----YY");  String s1 = sdf.format(date);  System.out.println(s1);  }  }  Q1685 |
| |  | | --- | | B.  Tue Dec 17 14:37:40 IST 2019  17----Dec----2019 |  |  | | --- | | C.  Tue Dec 17 14:37:40  17----Dec----2019 | | |

|  |  |
| --- | --- |
| **237.** | "package regularExpression;  import java.text.DateFormat;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  Date date = new Date();  System.out.println(date);  SimpleDateFormat sdf = new SimpleDateFormat(""dd----MMM----YYYY"");  String s1 = sdf.format(date);  System.out.println(s1);  }  }  "Q1686 |
| |  | | --- | | B.  Tue Dec 17 14:37:40 IST 2019  17----Dec----2019 |  |  | | --- | | C.  Tue Dec 17 14:37:40  17----Dec----2019 | | |
| **238.** | package regularExpression;  import java.text.DateFormat;  import java.text.NumberFormat;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class S7 {  public static void main(String[] args) {  double num = 4.56787887;  System.out.println(num);  NumberFormat nf = NumberFormat.getInstance();  String s1 = nf.format(num);  System.out.println(s1);  }  }  Q1687 |
| |  | | --- | | A.  4.56787887  4.568 |  |  | | --- | | B.  4.56787887  4.56877 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **239.** | package regularExpression;  import java.text.DateFormat;  import java.text.NumberFormat;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  import java.util.Locale;  public class S7 {  public static void main(String[] args) {  double num = 423654778954.56787887;  System.out.println(num);  NumberFormat nf = NumberFormat.getCurrencyInstance(Locale.UK);  String s1 = nf.format(num);  System.out.println(s1);  }  }  Q1688 |
| |  | | --- | | A.  4.236547789545679E11  $423,654,778,954.57 |  |  | | --- | | B.  4.236547789545679E11  £423,654,778,954.57 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | |
| **240.** | package regularExpression;  import java.text.DateFormat;  import java.text.NumberFormat;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  import java.util.Locale;  public class S7 {  public static void main(String[] args) {  Calendar cal = Calendar.getInstance();  Date date = cal.getTime();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.FULL, Locale.FRANCE);  String s1 = df.format(date);  System.out.println(s1);  }  }  Q1689 |
| |  | | --- | | A.  Tue Dec 17 14:49:54 IST 2019  mardi 17 décembre 2019 |  |  | | --- | | B.  Tue Dec 17 14:49:54 IST 2019  17 décembre 2019 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **241.** | package dateAndNumberFormats;  import java.util.Date;  public class S1 {  public static void main(String[] args) {  Date date = new Date();  System.out.println(date);  }  }  // what is the output dateQ1690 |
| |  | | --- | | A.  Today's date |  |  | | --- | | B.  Tomorrow's date |  |  | | --- | | C.  Yesterday's date |  |  | | --- | | D.  None | | |
| **242.** | package dateAndNumberFormats;  import java.text.DateFormat;  import java.text.ParseException;  import java.util.Date;  public class S2 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance();  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  }  Q1691 |
| |  | | --- | | A.  Tue Dec 17 15:24:16 IST 2019  Dec 17, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 15:24:16 IST 2019  17-Dec-2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **243.** | package dateAndNumberFormats;  import java.text.DateFormat;  import java.text.ParseException;  import java.util.Date;  public class S3 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.SHORT);  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  }  Q1692 |
| |  | | --- | | A.  Tue Dec 17 16:28:12 IST 2019  12/17/2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 16:28:12 IST 2019  Dec/17/19  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Tue Dec 17 16:28:12 IST 2019  12/17/19  Tue Dec 17 00:00:00 IST 2019 | | |
| **244.** | package dateAndNumberFormats;  import java.text.DateFormat;  import java.text.ParseException;  import java.util.Date;  public class S4 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.MEDIUM);  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  }  Q1693 |
| |  | | --- | | A.  Tue Dec 17 16:24:52 IST 2019  December 17, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 16:27:28 IST 2019  Dec 17, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **245.** | package dateAndNumberFormats;  import java.text.DateFormat;  import java.text.ParseException;  import java.util.Date;  public class S5 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.LONG);  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  }  Q1694 |
| |  | | --- | | A.  Tue Dec 17 16:24:52 IST 2019  December 17, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 16:24:52 IST 2019  17, December, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Compilation error |  |  | | --- | | D.  None | | |
| **246.** | package dateAndNumberFormats;  import java.text.DateFormat;  import java.text.ParseException;  import java.util.Date;  public class S6 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  DateFormat df = DateFormat.getDateInstance(DateFormat.FULL);  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  }  Q1695 |
| |  | | --- | | A.  Tue Dec 17 16:23:37 IST 2019  December 17, 2019, Tuesday  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 16:23:37 IST 2019  Tuesday, December 17, 2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **247.** | package dateAndNumberFormats;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Date;  public class S7 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  SimpleDateFormat df = new SimpleDateFormat("dd-MM-yyyy");  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  }  Q1696 |
| |  | | --- | | A.  Tue Dec 17 15:35:54 IST 2019  17-12-2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 15:35:54 IST 2019  17-Dec--2019  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Tue Dec 17 15:35:54 IST 2019  17-Dec--19Tue Dec 17 00:00:00 IST 2019 | | |
| **248.** | package dateAndNumberFormats;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Date;  public class S8 {  public static void main(String[] args) throws ParseException {  Date date = new Date();  System.out.println(date);  SimpleDateFormat df = new SimpleDateFormat("yyyy.MM.dd G 'at' HH:mm:ss z");  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  }  Q1697 |
| |  | | --- | | A.  Tue Dec 17 15:41:03 IST 2019  17.12.2019 AD at 15:41:03 IST  Tue Dec 17 15:41:03 IST 2019 |  |  | | --- | | B.  Tue Dec 17 15:41:03 IST 2019  2019.12.17 AD at 15:41:03 IST  Tue Dec 17 15:41:03 IST 2019 |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **249.** | package dateAndNumberFormats;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class S9 {  public static void main(String[] args) throws ParseException {  Calendar calci = Calendar.getInstance();  Date date = calci.getTime();  System.out.println(date);  SimpleDateFormat df = new SimpleDateFormat("h:mm a");  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  }  Q1698 |
| |  | | --- | | A.  Tue Dec 17 15:45:07 IST 2019  3:45:03 PM  Thu Jan 01 15:45:00 IST 1970 |  |  | | --- | | B.  Tue Dec 17 15:45:07 IST 2019  3:45 PM  Thu Jan 01 15:45:00 IST 1970 |  |  | | --- | | C.  Compilation error | | |
| **250.** | package dateAndNumberFormats;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class S10 {  public static void main(String[] args) throws ParseException {  Calendar calci = Calendar.getInstance();  calci.add(Calendar.DATE, 5);  Date date = calci.getTime();  System.out.println(date);  SimpleDateFormat df = new SimpleDateFormat("yyyyy.MMMMM.dd GGG hh:mm aaa");  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  }  Q1699 |
| |  | | --- | | A.  Compilation error |  |  | | --- | | B.  Sun Dec 22 15:49:52 IST 2019  2019.December.22 AD 03:49 PM  Sun Dec 22 15:49:00 IST 2019 |  |  | | --- | | C.  Sun Dec 22 15:49:52 IST 2019  02019.December.22 AD 03:49 PM  Sun Dec 22 15:49:00 IST 2019 | | |

|  |  |
| --- | --- |
| **251.** | package dateAndNumberFormats;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class S11 {  public static void main(String[] args) throws ParseException {  Calendar calci = Calendar.getInstance();  calci.add(Calendar.MONTH, -5);  Date date = calci.getTime();  System.out.println(date);  SimpleDateFormat df = new SimpleDateFormat("yyyyy.MM.dd'T'HH:mm:ss.SSSXXX");  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  }  Q1700 |
| |  | | --- | | A.  Wed Jul 17 15:55:24 IST 2019  02019.07.17T15:55:24.874+05:30  Wed Jul 17 15:55:24 IST 2019 |  |  | | --- | | B.  Wed Jul 17 15:55:24 IST 2019  02019.07.17T15:55:24.874+05:30:45  Wed Jul 17 15:55:24 IST 2019 |  |  | | --- | | C.  Compilation error | | |
| **252.** | package dateAndNumberFormats;  import java.text.NumberFormat;  import java.text.ParseException;  public class S12 {  public static void main(String[] args) throws ParseException {  double num1 = 385897.88888888;  System.out.println(num1);  NumberFormat nf = NumberFormat.getInstance();  String s1 = nf.format(num1);  double num2 = (Double) nf.parse(s1);  System.out.println(s1);  System.out.println(num2);  }  }  Q1701 |
| |  | | --- | | A.  385897.88888888  385,897.889  385,897.889 |  |  | | --- | | B.  385897.88888888  385,897.889  385897.889 |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **253.** | package dateAndNumberFormats;  import java.text.NumberFormat;  import java.text.ParseException;  public class S13 {  public static void main(String[] args) throws ParseException {  double num1 = 385897.88888888;  System.out.println(num1);  NumberFormat nf = NumberFormat.getCurrencyInstance();  String s1 = nf.format(num1);  double num2 = (Double) nf.parse(s1);  System.out.println(s1);  System.out.println(num2);  }  }  Q1702 |
| |  | | --- | | A.  385897.88888888  $385,897.89  385897.89 |  |  | | --- | | B.  385897.88888888  385 897,89 ¤  385897.89 |  |  | | --- | | C.  385897.88888888  £385,897.89  385897.89 |  |  | | --- | | D.  None | | |
| **254.** | package dateAndNumberFormats;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  import java.util.Locale;  public class S14 {  public static void main(String[] args) throws ParseException {  Calendar calci = Calendar.getInstance();  Date date = calci.getTime();  System.out.println(date);  SimpleDateFormat df = new SimpleDateFormat("EEEE MMMM dd yy", Locale.FRANCE);  String s1 = df.format(date);  System.out.println(s1);  Date d2 = df.parse(s1);  System.out.println(d2);  }  }  Q1703 |
| |  | | --- | | A.  Tue Dec 17 16:06:51 IST 2019  Tuesday December 17 19  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | B.  Tue Dec 17 16:06:12 IST 2019  mardi décembre 17 19  Tue Dec 17 00:00:00 IST 2019 |  |  | | --- | | C.  Tue Dec 17 16:08:25 IST 2019  martedì dicembre 17 19  Tue Dec 17 00:00:00 IST 2019 | | |

|  |  |
| --- | --- |
| **255.** | package dateAndNumberFormats;  import java.text.NumberFormat;  import java.text.ParseException;  import java.util.Locale;  public class S15 {  public static void main(String[] args) throws ParseException {  double num1 = 385897.88888888;  System.out.println(num1);  NumberFormat nf = NumberFormat.getInstance(Locale.ITALY);  String s1 = nf.format(num1);  double num2 = (Double) nf.parse(s1);  System.out.println(s1);  System.out.println(num2);  }  }  Q1704 |
| |  | | --- | | A.  385897.88888888  385,897.89  385,897.89 |  |  | | --- | | B.  385897.88888888  385,897.89  385897.89 |  |  | | --- | | C.  385897.88888888  385.897,889  385897.889 | | |
| **256.** | package app1;  import java.util.ArrayList;  public class M1 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(909);  list.add("abc");  list.add(true);  list.add('a');  System.out.println(list);  }  }  Q1947 |
| |  | | --- | | A.  [909, abc, a, true] |  |  | | --- | | B.  [909, abc, true, a] |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **257.** | package app1;  import java.util.ArrayList;  public class M2 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(909);  list.add(909);  list.add("abc");  list.add(true);  list.add('a');  System.out.println(list);  list.add(1, "vijay");  System.out.println(list);  }  }  Q1948 |
| |  | | --- | | A.  [909, 909, abc, true, a]  [909, vijay, 909, abc, true, a] |  |  | | --- | | B.  [909, 909, abc, true, a]  [909, vijay, abc, true, a] |  |  | | --- | | C.  Compilation error | | |
| **258.** | package app1;  import java.util.ArrayList;  public class M3 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(909);  list1.add(909);  list1.add("abc");  list1.add(true);  list1.add('a');  System.out.println(list1);  list1.set(1, "vijay");  System.out.println(list1);  }  }  Q1949 |
| |  | | --- | | A.  [909, 909, abc, true, a]  [909, vijay, 909, abc, true, a] |  |  | | --- | | B.  [909, 909, abc, true, a]  [909, vijay, abc, true, a] |  |  | | --- | | C.  Compilation error | | |

|  |  |
| --- | --- |
| **259.** | package app1;  import java.util.ArrayList;  public class M4 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(909);  list1.add(909);  list1.add("abc");  list1.add(true);  list1.add('a');  System.out.println(list1);  System.out.println(list1.contains(true));  System.out.println(list1.contains(90000));  }  }  Q1950 |
| |  | | --- | | A.  [909, 909, abc, true, a]  false  false |  |  | | --- | | B.  [909, 909, abc, true, a]  true  true |  |  | | --- | | C.  [909, 909, abc, true, a]  true  false | | |
| **260.** | package app1;  import java.util.ArrayList;  public class M5 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  for(String s1 : args) {  list1.add(s1);  }  System.out.println(list1);  }  }  // If you pass the command line arguments abc xyz xyz 123 500 500  what will be the outputQ1951 |
| |  | | --- | | A.  abc xyz xyz 123 500 500 |  |  | | --- | | B.  abc xyz 123 500 |  |  | | --- | | C.  [] | | |

|  |  |
| --- | --- |
| **261.** | package app1;  import java.util.ArrayList;  public class M6 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  for(String s1 : args) {  if(! list1.contains(s1))  list1.add(s1);  }  System.out.println(list1);  }  }  // If you pass the command line arguments abc xyz xyz 123 500 500  what will be the outputQ1952 |
| |  | | --- | | A.  abc xyz xyz 123 500 500 |  |  | | --- | | B.  abc xyz 123 500 |  |  | | --- | | C.  [] | | |
| **262.** | package app1;  import java.util.ArrayList;  import java.util.Scanner;  public class M7 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  Scanner sc = new Scanner(System.in);  do  {  System.out.println("Enter an element");  String s1 = sc.next();  if(! list1.contains(s1))  list1.add(s1);  System.out.println("You want to add one more (Y/N)?");  }while("Y".equalsIgnoreCase(sc.next()));  System.out.println(list1);  }  }  // If you give the inupt as  abc  xyz  xyz  123  500  500  what will be the outputQ1953 |
| |  | | --- | | A.  abc xyz xyz 123 500 500 |  |  | | --- | | B.  abc xyz 123 500 |  |  | | --- | | C.  [] | | |

|  |  |
| --- | --- |
| **263.** | package app1;  import java.util.ArrayList;  import java.util.Scanner;  public class M8 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  ArrayList list2 = new ArrayList(list1);  System.out.println(list2);  }  }  Q1954 |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  [90, 190, 910, 901, 290]  [90, 190, 910, 901, 290] | | |
| **264.** | package app1;  import java.util.ArrayList;  public class M9 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  ArrayList list2 = new ArrayList();  list2.add("abc");  list2.add(90000);  list2.addAll(list1);  System.out.println(list2);  }  }  Q1955 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  [ 90, 190, 910, 901, 290, abc, 90000] |  |  | | --- | | B.  [90, 190, 910, 901, 290]  [abc, 90000, 90, 190, 910, 901, 290] |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **265.** | package app1;  import java.util.ArrayList;  public class M11 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  Object obj = list1.get(3);  System.out.println(obj);  }  }  Q1956 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  901 |  |  | | --- | | B.  [90, 190, 910, 901, 290]  910 |  |  | | --- | | C.  [90, 190, 910, 901, 290]  IndexOutOfBoundsException | | |
| **266.** | package app1;  import java.util.ArrayList;  public class M12 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  int i = (Integer) list1.get(3);  System.out.println(i);  }  }  Q1957 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  901 |  |  | | --- | | B.  [90, 190, 910, 901, 290]  910 |  |  | | --- | | C.  [90, 190, 910, 901, 290]  IndexOutOfBoundsException | | |

|  |  |
| --- | --- |
| **267.** | package app1;  import java.util.ArrayList;  public class M13 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  int i = list1.size();  System.out.println(i);  }  }  Q1958 |
| |  | | --- | | A.  4 |  |  | | --- | | B.  5 | | |
| **268.** | package app1;  import java.util.ArrayList;  public class M14 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  int j = list1.size();  for(int i = 0; i < list1.size(); i++) {  System.out.println(list1.get(i));  }  System.out.println(j);  }  }  Q1959 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  90  190  910  901  290  5 |  |  | | --- | | B.  [90, 190, 910, 901, 290]  90 190 910 901 290  5 |  |  | | --- | | C.  5 | | |

|  |  |
| --- | --- |
| **269.** | package app1;  import java.util.ArrayList;  public class M15 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  int j = list1.size();  for(Object obj : list1) {  System.out.print(obj + ", ");  }  System.out.println();  System.out.println(j);  }  }  Q1960 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  90  190  910  901  290  5 |  |  | | --- | | B.  [90, 190, 910, 901, 290]  90 190 910 901 290  5 |  |  | | --- | | C.  5 | | |
| **270.** | package app1;  import java.util.ArrayList;  import java.util.Iterator;  public class M16 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  Iterator it = list1.iterator();  while(it.hasNext()) {  System.out.println(it.next());  }  }  }  Q1961 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  90  190  910  901  290 |  |  | | --- | | B.  [90, 190, 910, 901, 290]  90 190 910 901 290 |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **271.** | package app1;  import java.util.ArrayList;  import java.util.Iterator;  import java.util.ListIterator;  public class M17 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  ListIterator it = list1.listIterator();  while(it.hasNext()) {  System.out.println(it.next());  }  }  }  Q1962 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  90  190  910  901  290 |  |  | | --- | | B.  [90, 190, 910, 901, 290]  90 190 910 901 290 |  |  | | --- | | C.  Compilation Error | | |
| **272.** | package app1;  import java.util.ArrayList;  import java.util.ListIterator;  public class M18 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  Object obj = list1.remove(2);  System.out.println(list1);  System.out.println(obj);  }  }  Q1963 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  [90, 190, 901, 290] |  |  | | --- | | B.  [90, 190, 910, 901, 290]  [90, 190, 901, 290]  910 |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **273.** | package app1;  import java.util.ArrayList;  public class M19 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  boolean b1 = list1.remove(new Integer (901));  System.out.println(list1);  System.out.println(b1);  }  }  Q1964 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  [90, 190, 910, 290]  true |  |  | | --- | | B.  [90, 190, 910, 901, 290]  [90, 190, 910, 290] |  |  | | --- | | C.  Compilation Error | | |
| **274.** | package app1;  import java.util.ArrayList;  public class M20 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  boolean b1 = list1.remove(new Integer (9000000));  System.out.println(list1);  System.out.println(b1);  }  }  Q1965 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  [90, 190, 910, 901, 290]  true |  |  | | --- | | B.  [90, 190, 910, 901, 290]  [90, 190, 910, 901, 290]  false |  |  | | --- | | C.  [90, 190, 910, 901, 290]  [90, 190, 910, 901, 290] | | |

|  |  |
| --- | --- |
| **275.** | package app1;  import java.util.ArrayList;  public class M21 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);    ArrayList list2 = new ArrayList();  list2.add(90);  list2.add(190);  System.out.println(list2);  System.out.println("-----------------------");    list1.removeAll(list2);    System.out.println(list1);  System.out.println(list2);  }  }  Q1966 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  [90, 190]  -----------------------  [910, 901, 290]  [90, 190] |  |  | | --- | | B.  [90, 190, 910, 901, 290]  [90, 190]  -----------------------  [90, 190, 910, 901, 290]  [] |  |  | | --- | | C.  [90, 190, 910, 901, 290]  [90, 190]  -----------------------  []  [] | | |
| **276.** | package app1;  import java.util.ArrayList;  public class M22 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);    ArrayList list2 = new ArrayList();  list2.add(90);  list2.add(190);  System.out.println(list2);  System.out.println("-----------------------");    list1.retainAll(list2);    System.out.println(list1);  System.out.println(list2);  }  }  Q1967 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  [90, 190]  -----------------------  [90, 190, 910, 901, 290]  [90, 190] |  |  | | --- | | B.  [90, 190, 910, 901, 290]  [90, 190]  -----------------------  [910, 901, 290]  [90, 190] |  |  | | --- | | C.  [90, 190, 910, 901, 290]  [90, 190]  -----------------------  [90, 190]  [90, 190] | | |

|  |  |
| --- | --- |
| **277.** | package app1;  import java.util.ArrayList;  public class M23 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  list1.removeRange(1, 3);  System.out.println(list1);  }  }  Q1968 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  [90, 901, 290] |  |  | | --- | | B.  Compilation Error |  |  | | --- | | C.  Run time exception | | |
| **278.** | package app1;  import java.util.ArrayList;  public class M23 extends ArrayList {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  M23 list1 = new M23();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  System.out.println(list1);  list1.removeRange(1, 3);  System.out.println(list1);  }  }  Q1969 |
| |  | | --- | | A.  [90, 190, 910, 901, 290]  [90, 901, 290] |  |  | | --- | | B.  Compilation Error |  |  | | --- | | C.  Run time exception | | |

|  |  |
| --- | --- |
| **279.** | package app1;  import java.util.ArrayList;  import java.util.Collections;  public class M24 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  list1.add(500);  list1.add(140);  list1.add(400);  System.out.println(list1);  Collections.sort(list1);  System.out.println(list1);  }  }  Q1970 |
| |  | | --- | | A.  [90, 190, 910, 901, 290, 500, 140, 400]  [90, 140, 190, 290, 400, 500, 901, 910] |  |  | | --- | | B.  [90, 190, 910, 901, 290, 500, 140, 400]  [910, 901, 500, 400, 290, 190, 140, 90] |  |  | | --- | | C.  Compilation Error | | |
| **280.** | package app1;  import java.util.ArrayList;  import java.util.Collections;  public class M25 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  ArrayList list1 = new ArrayList();  list1.add(90);  list1.add(190);  list1.add(910);  list1.add(901);  list1.add(290);  list1.add(500);  list1.add(140);  list1.add(400);  System.out.println(list1);  Collections.sort(list1);  System.out.println(list1);  int i = Collections.binarySearch(list1, 910);  System.out.println(i);  }  }  Q1971 |
| |  | | --- | | A.  [90, 190, 910, 901, 290, 500, 140, 400]  [90, 140, 190, 290, 400, 500, 901, 910]  910 |  |  | | --- | | B.  [90, 190, 910, 901, 290, 500, 140, 400]  [90, 140, 190, 290, 400, 500, 901, 910]  7 |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **281.** | entire collection api is available inQ1972 |
| |  | | --- | | A.  java.util |  |  | | --- | | B.  java.lang |  |  | | --- | | C.  java.awt | | |
| **282.** | Collection objects are fixed in sizeQ1973 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |

|  |  |
| --- | --- |
| **283.** | Arrays are fixed in sizeQ1974 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |
| **284.** | In collection api we can add any type of elementsQ1975 |
| |  | | --- | | A.  True |  |  | | --- | | B.  False | | |

|  |  |
| --- | --- |
| **285.** | Collection api allows to store only objectsQ1976 |
| |  | | --- | | A.  True |  |  | | --- | | B.  False | | |
| **286.** | Is it possible to store Wrapper class objectsQ1977 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |

|  |  |
| --- | --- |
| **287.** | By default most of the collections accept duplicatesQ1978 |
| |  | | --- | | A.  True |  |  | | --- | | B.  False | | |
| **288.** | Is it possible to copy one ArrayList elements into another ArrayList elementsQ1979 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |

|  |  |
| --- | --- |
| **289.** | By using which initializer you will copy one ArrayList elements into another ArrayListQ1980 |
| |  | | --- | | A.  SIB |  |  | | --- | | B.  constructor |  |  | | --- | | C.  IIB | | |
| **290.** | Is it possible to read collection objects through iteratorQ1981 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |

|  |  |
| --- | --- |
| **291.** | Is it possible to use removeRange() method stright awayQ1982 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |
| **292.** | In which collection type order of elements is preservedQ1983 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Queue |  |  | | --- | | C.  Map | | |

|  |  |
| --- | --- |
| **293.** | Which collection type does not allows duplicatesQ1984 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Queue |  |  | | --- | | C.  Map |  |  | | --- | | D.  Set | | |
| **294.** | In order to maintain elements in Queue or stack format which collection type is usedQ1985 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Queue |  |  | | --- | | C.  Map |  |  | | --- | | D.  Set | | |

|  |  |
| --- | --- |
| **295.** | In which collection type elements having both key value pairQ1986 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Queue |  |  | | --- | | C.  Map |  |  | | --- | | D.  Set | | |
| **296.** | In which collection type elements stored without index or keyQ1987 |
| |  | | --- | | A.  List |  |  | | --- | | B.  Queue |  |  | | --- | | C.  Map |  |  | | --- | | D.  Set | | |

|  |  |
| --- | --- |
| **297.** | Collection is anQ1988 |
| |  | | --- | | A.  class |  |  | | --- | | B.  interface | | |
| **298.** | Which class acts as both List and Queue typesQ1989 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  Vector |  |  | | --- | | C.  LinkedList | | |

|  |  |
| --- | --- |
| **299.** | Which class can be used in multi threaded environmentQ1990 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  Vector |  |  | | --- | | C.  LinkedList | | |
| **300.** | which class is synchronizedQ1991 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  Vector |  |  | | --- | | C.  LinkedList | | |

|  |  |
| --- | --- |
| **301.** | Which class is very much effective in case of reading operationQ1992 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  Vector |  |  | | --- | | C.  LinkedList | | |
| **302.** | ArrayList internally using arraysQ1993 |
| |  | | --- | | A.  True |  |  | | --- | | B.  False | | |

|  |  |
| --- | --- |
| **303.** | Which class is more preferable for in case of more frequently reading operationQ1994 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  Vector |  |  | | --- | | C.  LinkedList | | |
| **304.** | Which class is not much effective in case of a reading operationQ1995 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  Vector |  |  | | --- | | C.  LinkedList | | |

|  |  |
| --- | --- |
| **305.** | In which class insertion operation is not advisableQ1996 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  Vector |  |  | | --- | | C.  LinkedList | | |
| **306.** | In which class insertion operations are more effectiveQ1997 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  Vector |  |  | | --- | | C.  LinkedList | | |

|  |  |
| --- | --- |
| **307.** | Which class having dual featuresQ1998 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  Vector |  |  | | --- | | C.  LinkedList | | |
| **308.** | Which class is a legacy classQ1999 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  Vector |  |  | | --- | | C.  LinkedList | | |

|  |  |
| --- | --- |
| **309.** | which class is non synchronized by defaultQ2000 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  Vector | | |
| **310.** | Which class is used for sorting purposeQ2001 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  Vector |  |  | | --- | | C.  LinkedList |  |  | | --- | | C.  Collections | | |

|  |  |
| --- | --- |
| **311.** | package queue;  import java.util.LinkedList;  public class M1 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  LinkedList list = new LinkedList();  list.add(90);  list.add(100);  list.add(1190);  list.add(1910);  list.add(1901);  list.add(1290);  System.out.println(list);  System.out.println(list.removeFirst());  System.out.println(list);  }  }  Q2002 |
| |  | | --- | | A.  [90, 100, 1190, 1910, 1901, 1290]  90  [100, 1190, 1910, 1901, 1290] |  |  | | --- | | B.  [90, 100, 1190, 1910, 1901, 1290]  [100, 1190, 1910, 1901, 1290] | | |
| **312.** | package queue;  import java.util.LinkedList;  public class M2 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  LinkedList list = new LinkedList();  list.add(90);  list.add(100);  list.add(1190);  list.add(1910);  list.add(1901);  list.add(1290);  System.out.println(list);  System.out.println(list.removeLast());  System.out.println(list);  System.out.println(list.removeLast());  System.out.println(list);  }  }  Q2003 |
| |  | | --- | | A.  [90, 100, 1190, 1910, 1901, 1290]  [90, 100, 1190, 1910, 1901]  [90, 100, 1190, 1910] |  |  | | --- | | B.  [90, 100, 1190, 1910, 1901, 1290]  1290  [90, 100, 1190, 1910, 1901]  1901  [90, 100, 1190, 1910] |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **313.** | package queue;  import java.util.LinkedList;  public class M3 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  LinkedList list = new LinkedList();  list.add(90);  list.add(100);  list.add(1190);  list.add(1910);  list.add(1901);  list.add(1290);  System.out.println(list);  System.out.println(list.peek());  System.out.println(list);  System.out.println(list.peek());  System.out.println(list);  }  }  Q2004 |
| |  | | --- | | A.  [90, 100, 1190, 1910, 1901, 1290]  [90, 100, 1190, 1910, 1901, 1290]  [90, 100, 1190, 1910, 1901, 1290] |  |  | | --- | | B.  [90, 100, 1190, 1910, 1901, 1290]  1290  [90, 100, 1190, 1910, 1901, 1290]  1290  [90, 100, 1190, 1910, 1901, 1290] |  |  | | --- | | C.  [90, 100, 1190, 1910, 1901, 1290]  90  [90, 100, 1190, 1910, 1901, 1290]  90  [90, 100, 1190, 1910, 1901, 1290] | | |
| **314.** | package queue;  import java.util.LinkedList;  public class M4 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  LinkedList list = new LinkedList();  list.add(90);  list.add(100);  list.add(1190);  list.add(1910);  list.add(1901);  list.add(1290);  System.out.println(list);  System.out.println(list.poll());  System.out.println(list);  System.out.println(list.poll());  System.out.println(list);  }  }  Q2005 |
| |  | | --- | | A.  [90, 100, 1190, 1910, 1901, 1290]  90  [100, 1190, 1910, 1901, 1290]  1290  [1190, 1910, 1901] |  |  | | --- | | B.  [90, 100, 1190, 1910, 1901, 1290]  [100, 1190, 1910, 1901, 1290]  [1190, 1910, 1901, 1290] |  |  | | --- | | C.  [90, 100, 1190, 1910, 1901, 1290]  90  [100, 1190, 1910, 1901, 1290]  100  [1190, 1910, 1901, 1290] | | |

|  |  |
| --- | --- |
| **315.** | package queue;  import java.util.LinkedList;  public class M5 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  LinkedList list = new LinkedList();  list.add(90);  list.add(100);  list.add(1190);  list.add(1910);  list.add(1901);  list.add(1290);  System.out.println(list);  System.out.println(list.peekLast());  System.out.println(list);  System.out.println(list.peekLast());  System.out.println(list);  }  }  Q2006 |
| |  | | --- | | A.  [90, 100, 1190, 1910, 1901, 1290]  [90, 100, 1190, 1910, 1901, 1290]  [90, 100, 1190, 1910, 1901, 1290] |  |  | | --- | | B.  [90, 100, 1190, 1910, 1901, 1290]  1290  [90, 100, 1190, 1910, 1901, 1290]  1290  [90, 100, 1190, 1910, 1901, 1290] |  |  | | --- | | C.  [90, 100, 1190, 1910, 1901, 1290]  90  [90, 100, 1190, 1910, 1901, 1290]  90  [90, 100, 1190, 1910, 1901, 1290] | | |
| **316.** | package queue;  import java.util.LinkedList;  public class M6 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  LinkedList list = new LinkedList();  list.add(90);  list.add(100);  list.add(1190);  list.add(1910);  list.add(1901);  list.add(1290);  System.out.println(list);  System.out.println(list.pollLast());  System.out.println(list);  System.out.println(list.pollLast());  System.out.println(list);  }  }  Q2007 |
| |  | | --- | | A.  [90, 100, 1190, 1910, 1901, 1290]  90  [100, 1190, 1910, 1901, 1290]  1290  [1190, 1910, 1901] |  |  | | --- | | B.  [90, 100, 1190, 1910, 1901, 1290]  1290  [100, 1190, 1910, 1901]  1901  [1190, 1910, 1901, 1290] |  |  | | --- | | C.  [90, 100, 1190, 1910, 1901, 1290]  90  [100, 1190, 1910, 1901, 1290]  100  [1190, 1910, 1901, 1290] | | |

|  |  |
| --- | --- |
| **317.** | package queue;  import java.util.PriorityQueue;;  public class M7 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  PriorityQueue queue = new PriorityQueue();  queue.add(90);  queue.add(100);  queue.add(1190);  queue.add(1910);  queue.add(1901);  queue.add(1290);  queue.add(10);  System.out.println(queue);  System.out.println(queue.peek());  System.out.println(queue);  System.out.println(queue.peek());  System.out.println(queue);  }  }  Q2008 |
| |  | | --- | | A.  [10, 100, 90, 1910, 1901, 1290, 1190]  10  [10, 100, 90, 1910, 1901, 1290, 1190]  10  [10, 100, 90, 1910, 1901, 1290, 1190] |  |  | | --- | | B.  [100, 90, 1910, 1901, 1290, 1190, 10]  100  [100, 90, 1910, 1901, 1290, 1190, 10]  100  [100, 90, 1910, 1901, 1290, 1190, 10] |  |  | | --- | | C.  None | | |
| **318.** | package queue;  import java.util.PriorityQueue;  public class M8 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  PriorityQueue queue = new PriorityQueue();  queue.add(90);  queue.add(100);  queue.add(1190);  queue.add(1910);  queue.add(1901);  queue.add(1290);  queue.add(10);  System.out.println(queue);  System.out.println(queue.poll());  System.out.println(queue);  System.out.println(queue.poll());  System.out.println(queue);  }  }  Q2009 |
| |  | | --- | | A.  [10, 100, 90, 1910, 1901, 1290, 1190]  10  [10, 100, 90, 1910, 1901, 1290, 1190]  10  [10, 100, 90, 1910, 1901, 1290, 1190] |  |  | | --- | | B.  [100, 90, 1910, 1901, 1290, 1190, 10]  100  [100, 90, 1910, 1901, 1290, 1190, 10]  100  [100, 90, 1910, 1901, 1290, 1190, 10] |  |  | | --- | | C.  [10, 100, 90, 1910, 1901, 1290, 1190]  10  [90, 100, 1190, 1910, 1901, 1290]  90  [100, 1290, 1190, 1910, 1901] | | |

|  |  |
| --- | --- |
| **319.** | package queue;  import java.util.PriorityQueue;  public class M9 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  PriorityQueue queue = new PriorityQueue();  queue.add(90);  queue.add(100);  queue.add(1190);  System.out.println(queue.get());  }  }  Q2010 |
| |  | | --- | | A.  90  100  1190 |  |  | | --- | | B.  90 100 1190 |  |  | | --- | | C.  Compilation Error | | |
| **320.** | package queue;  import java.util.PriorityQueue;  public class M9 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  PriorityQueue queue = new PriorityQueue();  queue.add(90);  queue.add(100);  queue.add(1190);  for(Object obj : queue) {  System.out.println(obj);  }  }  }  Q2011 |
| |  | | --- | | A.  90  100  1190 |  |  | | --- | | B.  90 100 1190 |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **321.** | package queue;  import java.util.PriorityQueue;  public class M10 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  PriorityQueue queue = new PriorityQueue();  queue.add(90);  queue.add(100.0);  queue.add(1190);  queue.add(119000);  System.out.println(queue);  }  }  Q2012 |
| |  | | --- | | A.  90  100.0  1190  119000 |  |  | | --- | | B.  90 100 1190 119000 |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  ClassCastException | | |
| **322.** | package queue;  import java.util.PriorityQueue;  public class M11 {  @SuppressWarnings("unchecked")  public static void main(String[] args) {  PriorityQueue queue = new PriorityQueue();  queue.add(90);  queue.add(null);  queue.add(1190);  queue.add(119000);  System.out.println(queue);  }  }  Q2013 |
| |  | | --- | | A.  90  null  1190  119000 |  |  | | --- | | B.  90 null 1190 119000 |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  NullPointerException | | |

|  |  |
| --- | --- |
| **323.** | Where ever queue is possible there and all stack also possibleQ2014 |
| |  | | --- | | A.  True |  |  | | --- | | B.  False | | |
| **324.** | We are achiving perfect stack with the help of poll() methodQ2015 |
| |  | | --- | | A.  True |  |  | | --- | | B.  False | | |

|  |  |
| --- | --- |
| **325.** | Which class having default auto sortingQ2016 |
| |  | | --- | | A.  LinkedList |  |  | | --- | | B.  PriorityQueue | | |
| **326.** | Is it possible to read elements in the PriorityQueue by using get() methodQ2017 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |

|  |  |
| --- | --- |
| **327.** | Is it possible to use enhanced loop to read the elements in LinkedListQ2018 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |
| **328.** | We can insert any type of elements in PriorityQueueQ2019 |
| |  | | --- | | A.  True |  |  | | --- | | B.  False | | |

|  |  |
| --- | --- |
| **329.** | Is it possible to supply null elements to priorityQueueQ2020 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |
| **330.** | package set;  import java.util.HashSet;  class A  {  int i;  A(int i){  this.i = i;  }  public String toString() {  return "(i = " + i + ")";  }  }  public class M1 {  public static void main(String[] args) {  HashSet set = new HashSet();  set.add(new A(90));  set.add(new A(90));  set.add(new A(90));  set.add(new A(90));  System.out.println(set);  }  }  Q2021 |
| |  | | --- | | A.  [(i = 90), (i = 90), (i = 90), (i = 90)] |  |  | | --- | | B.  [(i = 90)] |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **331.** | package set;  import java.util.HashSet;  class A  {  int i;  A(int i){  this.i = i;  }  public String toString() {  return "(i = " + i + ")";  }  @Override  public int hashCode() {  return i;  }  public boolean equals(Object obj) {  return i == ((A)obj).i;  }  }  public class M1 {  public static void main(String[] args) {  HashSet set = new HashSet();  set.add(new A(90));  set.add(new A(90));  set.add(new A(90));  set.add(new A(90));  System.out.println(set);  }  }  Q2022 |
| |  | | --- | | A.  [(i = 90), (i = 90), (i = 90), (i = 90)] |  |  | | --- | | B.  [(i = 90)] |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |
| **332.** | package set;  import java.util.HashSet;  public class M2 {  public static void main(String[] args) {  HashSet set = new HashSet();  set.add(90);  set.add(90);  set.add(90);  set.add(90);  set.add(90);  set.add(90);  System.out.println(set);  }  }  Q2023 |
| |  | | --- | | A.  [90, 90, 90, 90, 90,90] |  |  | | --- | | B.  [90] |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **333.** | package set;  import java.util.HashSet;  public class M3 {  public static void main(String[] args) {  HashSet set = new HashSet();  System.out.println(set.add(90));  System.out.println(set.add(90));  System.out.println(set.add(90));  System.out.println(set.add(90));  System.out.println(set.add(90));  System.out.println(set);  }  }  Q2024 |
| |  | | --- | | A.  true  false  false  false  false  [90] |  |  | | --- | | B.  false  true  true  true  true  true  [90] |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |
| **334.** | package set;  import java.util.HashSet;  public class M3 {  public static void main(String[] args) {  HashSet set = new HashSet();  System.out.println(set.add(90));  System.out.println(set.add(90));  System.out.println(set.add(90));  System.out.println(set.add(90));  System.out.println(set.add(91));  System.out.println(set);  }  }  Q2025 |
| |  | | --- | | A.  true  false  false  false  false  [90, 91] |  |  | | --- | | B.  false  true  true  true  true  true  [90, 91] |  |  | | --- | | C.  true  false  false  false  true  [90, 91] |  |  | | --- | | D.  Compilation Error | | |

|  |  |
| --- | --- |
| **335.** | package set;  import java.util.HashSet;  public class M4 {  public static void main(String[] args) {  HashSet set = new HashSet();  System.out.println(set.add("hello"));  System.out.println(set.add("hello"));  System.out.println(set.add("abc"));  System.out.println(set.add("abc"));  System.out.println(set.add("xyz"));  System.out.println(set.add("xyz"));  System.out.println(set);  }  }  Q2026 |
| |  | | --- | | A.  true  false  true  false  false  false  [abc, xyz, hello] |  |  | | --- | | B.  true  false  true  false  true  false  [abc, xyz, hello] |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |
| **336.** | package set;  import java.util.HashSet;  public class M5 {  public static void main(String[] args) {  HashSet set = new HashSet();  System.out.println(set.add(new StringBuffer("hello")));  System.out.println(set.add(new StringBuffer("hello")));  System.out.println(set.add(new StringBuffer("abc")));  System.out.println(set.add(new StringBuffer("abc")));  System.out.println(set.add(new StringBuffer("xyz")));  System.out.println(set.add(new StringBuffer("xyz")));  System.out.println(set);  }  }  Q2027 |
| |  | | --- | | A.  true  false  true  false  true  false  [abc, xyz, hello] |  |  | | --- | | B.  Compilation Error |  |  | | --- | | C.  true  true  true  true  true  true  [abc, xyz, xyz, hello, hello, abc] | | |

|  |  |
| --- | --- |
| **337.** | package set;  import java.util.HashSet;  class B{  int i, j;  B(int i, int j){  this.i = i;  this.j = j;  }  public String toString() {  return "(" + i + ", " + j + ")";  }  public int hashCode() {  String s1 = Integer.toString(i);  String s2 = Integer.toString(j);  int hash = s1.hashCode();  hash += s2.hashCode();  return hash;  }  public boolean equals(Object obj) {  return (i == ((B)obj).i && j == ((B)obj).j);  }  }  public class M6 {  public static void main(String[] args) {  HashSet set = new HashSet();  System.out.println(set.add(new B(90,10)));  System.out.println(set.add(new B(90,10)));  System.out.println(set.add(new B(90,10)));  System.out.println(set.add(new B(90,10)));  System.out.println(set.add(new B(90,11)));  System.out.println(set.add(new B(90,11)));  System.out.println(set.add(new B(90,11)));  System.out.println(set.add(new B(90,10)));  System.out.println(set.add(new B(90,10)));  System.out.println(set.add(new B(90,12)));  System.out.println(set);  }  }  Q2028 |
| |  | | --- | | A.  true  false  false  false  true  false  false  false  false  true  [(90, 10), (90, 11), (90, 12)] |  |  | | --- | | B.  true  false  false  false  true  false  false  false  false  false  [(90, 10), (90, 11)] |  |  | | --- | | C.  Compilation Error | | |
| **338.** | package set;  import java.util.HashSet;  class D  {  int i;  D(int i){  this.i = i;  }  public String toString() {  return "(" + i + ")";  }  }  public class M8 {  public static void main(String[] args) {  D d1 = new D(90);  D d2 = new D(90);  HashSet set = new HashSet();  set.add(d1);  set.add(d2);  System.out.println(set);  }  }  Q2029 |
| |  | | --- | | A.  [(90)] |  |  | | --- | | B.  [(90), (90)] |  |  | | --- | | C.  None | | |

|  |  |
| --- | --- |
| **339.** | package set;  import java.util.HashSet;  class E  {  int i;  E(int i){  this.i = i;  }  public String toString() {  return "(" + i + ")";  }  public int hashCode() {  String s1 = Integer.toString(i);  int hash = s1.hashCode();  return hash;  }  public boolean equals(Object obj) {  boolean b1 = (i == ((E)obj).i);  return b1;  }  }  public class M9 {  public static void main(String[] args) {  E d1 = new E(90);  E d2 = new E(90);  HashSet set = new HashSet();  set.add(d1);  set.add(d2);  System.out.println(set);  }  }  Q2030 |
| |  | | --- | | A.  [(90)] |  |  | | --- | | B.  [(90), (90)] |  |  | | --- | | C.  None | | |
| **340.** | package set;  import java.util.LinkedHashSet;  public class M11 {  public static void main(String[] args) {  LinkedHashSet set = new LinkedHashSet();  set.add(90);  set.add(90);  set.add(190);  set.add(190);  set.add(90);  set.add(190);  set.add(290);  set.add(390);  set.add(390);  set.add(290);  set.add(390);  set.add(500);  System.out.println(set);  }  }  Q2031 |
| |  | | --- | | A.  [90, 290, 190, 500,390] |  |  | | --- | | B.  [90, 190, 290, 390, 500] |  |  | | --- | | C.  None | | |

|  |  |
| --- | --- |
| **341.** | Is it possible to sort elements under TreeSetQ2032 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |
| **342.** | package set;  import java.util.HashSet;  import java.util.TreeSet;  public class M12 {  public static void main(String[] args) {  HashSet set = new HashSet();  set.add(90);  set.add(190);  set.add(910);  set.add(901);  set.add(290);  set.add(920);  set.add(500);  System.out.println(set);  TreeSet set1 = new TreeSet();  set1.addAll(set);  System.out.println(set1);  }  }  Q2033 |
| |  | | --- | | A.  [290, 500, 901, 920, 90, 190, 910]  [90, 190, 290, 500, 901, 910, 920] |  |  | | --- | | B.  [290, 500, 901, 920, 90, 190, 910]  [90, 190, 290, 500, 920, 901, 910] |  |  | | --- | | C.  None | | |

|  |  |
| --- | --- |
| **343.** | package map;  import java.util.HashMap;  public class M1 {  public static void main(String[] args) {  HashMap map1 = new HashMap();  map1.put("key1", 2314);  map1.put(345, "rty");  map1.put(3.4, true);  map1.put('a', false);  System.out.println(map1);  }  }  Q2034 |
| |  | | --- | | A.  {key1=2314, a=false, 3.4=true, 345=rty} |  |  | | --- | | B.  {key1=2314, 345=rty, 3.4=true, a=false} | | |
| **344.** | package map;  import java.util.HashMap;  public class M2 {  public static void main(String[] args) {  HashMap map1 = new HashMap();  map1.put(null, 2314);  map1.put(345, "rty");  map1.put(3.4, true);  map1.put('a', null);  System.out.println(map1);  }  }  Q2035 |
| |  | | --- | | A.  NullPointerException |  |  | | --- | | B.  Compilation Error |  |  | | --- | | C.  {null=2314, a=null, 3.4=true, 345=rty} | | |

|  |  |
| --- | --- |
| **345.** | package map;  import java.util.HashMap;  public class M3 {  public static void main(String[] args) {  HashMap map1 = new HashMap();  map1.put(null, 2314);  map1.put(345, "rty");  map1.put(3.4, true);  map1.put('a', null);  System.out.println(map1);  String s1 = (String) map1.get(345);  System.out.println(s1);  }  }  Q2036 |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  {null=2314, a=null, 3.4=true, 345=rty}  rty |  |  | | --- | | C.  {null=2314, a=null, 3.4=true, 345=rty}  345 | | |
| **346.** | package map;  import java.util.HashMap;  public class M3 {  public static void main(String[] args) {  HashMap map1 = new HashMap();  map1.put(null, 2314);  map1.put(345, "rty");  map1.put(3.4, true);  map1.put('a', null);  System.out.println(map1);  String s1 = (String) map1.get(345);  System.out.println(s1);  String s2 = (String) map1.get("xyz");  System.out.println(s2);  }  }  Q2037 |
| |  | | --- | | A.  {null=2314, a=null, 3.4=true, 345=rty}  rty |  |  | | --- | | B.  NullPointerException | | |

|  |  |
| --- | --- |
| **347.** | package map;  import java.util.HashMap;  import java.util.Set;  public class M4 {  public static void main(String[] args) {  HashMap map1 = new HashMap();  map1.put("abc", 2314);  map1.put(345, "rty");  map1.put(3.4, true);  map1.put('a', "xyz");    Set keys = map1.keySet();  System.out.println(keys);  }  }  Q2038 |
| |  | | --- | | A.  [a, abc, 3.4, 345] |  |  | | --- | | B.  Compilation Error | | |
| **348.** | package map;  import java.util.HashMap;  import java.util.Set;  public class M5 {  public static void main(String[] args) {  HashMap map1 = new HashMap();  map1.put("abc", 2314);  map1.put(345, "rty");  map1.put(3.4, true);  map1.put('a', "xyz");  Set keys = map1.keySet();  for(Object key : keys) {  System.out.println(key + ":" + map1.get(key));  }  System.out.println(keys);  }  }  Q2039 |
| |  | | --- | | A.  a  abc  3.4  345  [a, abc, 3.4, 345] |  |  | | --- | | B.  a:xyz  abc:2314  3.4:true  345:rty  [a, abc, 3.4, 345] |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **349.** | package map;  import java.util.HashMap;  import java.util.Set;  public class M6 {  public static void main(String[] args) {  HashMap map1 = new HashMap();  map1.put("abc", 2314);  map1.put(345, "rty");  map1.put(3.4, true);  map1.put('a', "xyz");  Set entries = map1.entrySet();  for(Object entry : entries) {  System.out.println(entry);  }  }  }  Q2040 |
| |  | | --- | | A.  xyz  2314  true  rty |  |  | | --- | | B.  a=xyz  abc=2314  3.4=true  345=rty |  |  | | --- | | C.  Compilation Error | | |
| **350.** | package map;  import java.util.HashMap;  import java.util.Map.Entry;  import java.util.Set;  public class M7 {  public static void main(String[] args) {  HashMap map1 = new HashMap();  map1.put("abc", 2314);  map1.put(345, "rty");  map1.put(3.4, true);  map1.put('a', "xyz");  Set entries = map1.entrySet();  Entry entry;  for(Object obj : entries) {  entry = (Entry) obj;  System.out.println(entry.getKey() + ":" + entry.getValue());  }  }  }  Q2041 |
| |  | | --- | | A.  xyz  2314  true  rty |  |  | | --- | | B.  a=xyz  abc=2314  3.4=true  345=rty |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **351.** | package map;  import java.util.Hashtable;  public class M9 {  public static void main(String[] args) {  Hashtable table = new Hashtable();  table.put("abc", 3456);  table.put("a", 345);  table.put(null, true);  table.put(true, 3.4);  System.out.println(table);  }  }  Q2042 |
| |  | | --- | | A.  abc=3456  a=345  null=true  true=3.4 |  |  | | --- | | B.  Compilation Error |  |  | | --- | | C.  NullPointerExxeption | | |
| **352.** | package map;  import java.util.LinkedHashMap;  public class M11 {  public static void main(String[] args) {  LinkedHashMap map = new LinkedHashMap();  map.put("abc", 34);  map.put("abc1", 34);  map.put("abc2", 34);  map.put("abc3", 34);  map.put("abc4", 34);  map.put("abc5", 34);  map.put("abc6", 34);  map.put("abc7", 34);  System.out.println(map);  }  }  Q2043 |
| |  | | --- | | A.  {abc=34, abc1=34, abc2=34, abc3=34, abc4=34, abc5=34, abc6=34, abc7=34} |  |  | | --- | | B.  {abc=34, abc1=34, abc3=34, abc2=34, abc4=34, abc7=34, abc6=34, abc5=34} |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **353.** | package map;  import java.util.HashMap;  import java.util.TreeMap;  class A implements Comparable{  int i;  A(int i){  this.i = i;  }  public String toString() {  return "(i = " + i + ")";  }  public int compareTo(Object obj) {  return i - ((A)obj).i;  }  }  public class M13 {  public static void main(String[] args) {  HashMap map = new HashMap();  map.put(new A(90), "test1");  map.put(new A(9), "test2");  map.put(new A(0), "test3");  map.put(new A(190), "test4");  map.put(new A(910), "test5");  map.put(new A(901), "test6");  System.out.println(map);  TreeMap map1 = new TreeMap();  map1.putAll(map);  System.out.println(map1);  }  }  Q2044 |
| |  | | --- | | A.  {(i = 0)=test3, (i = 901)=test6, (i = 910)=test5, (i = 90)=test1, (i = 9)=test2, (i = 190)=test4}  {(i = 0)=test3, (i = 9)=test2, (i = 90)=test1, (i = 190)=test4, (i = 901)=test6, (i = 910)=test5} |  |  | | --- | | B.  {(i = 0)=test3, (i = 9)=test2, (i = 90)=test1, (i = 190)=test4, (i = 901)=test6, (i = 910)=test5}  {(i = 0)=test3, (i = 901)=test6, (i = 910)=test5, (i = 90)=test1, (i = 9)=test2, (i = 190)=test4} |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |
| **354.** | package map;  import java.util.Comparator;  import java.util.HashMap;  import java.util.TreeMap;  class B {  int i, j;  B(int i, int j){  this.i = i;  this.j = j;  }  public String toString() {  return "(" + i + "," + j + ")";  }  }  class SortBasedOnI implements Comparator{  public int compare(Object obj1, Object obj2) {  return ((B)obj1).i - ((B)obj2).i;  }  }  class SortBasedOnJ implements Comparator{  public int compare(Object obj1, Object obj2) {  return ((B)obj1).j - ((B)obj2).j;  }  }  public class M14 {  public static void main(String[] args) {  HashMap map = new HashMap();  map.put(new B(90, 0), "test1");  map.put(new B(9, 56), "test2");  map.put(new B(0, 125), "test3");  map.put(new B(190, 45), "test4");  map.put(new B(910, 50), "test5");  map.put(new B(901, 52), "test6");  System.out.println(map);  TreeMap map1 = new TreeMap(new SortBasedOnI());  map1.putAll(map);  System.out.println(map1);    TreeMap map2 = new TreeMap(new SortBasedOnJ());  map2.putAll(map);  System.out.println(map2);  }  }  Q2045 |
| |  | | --- | | A.  {(0,125)=test3, (901,52)=test6, (910,50)=test5, (90,0)=test1, (9,56)=test2, (190,45)=test4}  {(90,0)=test1, (190,45)=test4, (910,50)=test5, (901,52)=test6, (9,56)=test2, (0,125)=test3}  {(0,125)=test3, (9,56)=test2, (90,0)=test1, (190,45)=test4, (901,52)=test6, (910,50)=test5} |  |  | | --- | | B.  {(0,125)=test3, (901,52)=test6, (910,50)=test5, (90,0)=test1, (9,56)=test2, (190,45)=test4}  {(0,125)=test3, (9,56)=test2, (90,0)=test1, (190,45)=test4, (901,52)=test6, (910,50)=test5}  {(90,0)=test1, (190,45)=test4, (910,50)=test5, (901,52)=test6, (9,56)=test2, (0,125)=test3} | | |

|  |  |
| --- | --- |
| **355.** | Which class is used for sorting in list type elementsQ2046 |
| |  | | --- | | A.  Collections class |  |  | | --- | | B.  Priority class |  |  | | --- | | C.  TreeSet |  |  | | --- | | D.  TreeMap |  |  | | --- | | E.  None | | |
| **356.** | Which class is used for sorting in Queue type elementsQ2047 |
| |  | | --- | | A.  Collections class |  |  | | --- | | B.  Priority class |  |  | | --- | | C.  TreeSet |  |  | | --- | | D.  TreeMap |  |  | | --- | | E.  None | | |

|  |  |
| --- | --- |
| **357.** | Which class is used for sorting in set type elementsQ2048 |
| |  | | --- | | A.  Collections class |  |  | | --- | | B.  Priority class |  |  | | --- | | C.  TreeSet |  |  | | --- | | D.  TreeMap |  |  | | --- | | E.  None | | |
| **358.** | Which class is used for sorting in Map type elementsQ2049 |
| |  | | --- | | A.  Collections class |  |  | | --- | | B.  Priority class |  |  | | --- | | C.  TreeSet |  |  | | --- | | D.  TreeMap |  |  | | --- | | E.  None | | |

|  |  |
| --- | --- |
| **359.** | Which class is non-Synchronized oneQ2050 |
| |  | | --- | | A.  HashMap |  |  | | --- | | B.  Hashtable |  |  | | --- | | C.  None | | |
| **360.** | Which one is synchronized oneQ2051 |
| |  | | --- | | A.  HashMap |  |  | | --- | | B.  Hashtable |  |  | | --- | | C.  None | | |

|  |  |
| --- | --- |
| **361.** | Which class allows null as key value pairQ2052 |
| |  | | --- | | A.  HashMap |  |  | | --- | | B.  Hashtable |  |  | | --- | | C.  None | | |
| **362.** | Which class does not allow null as key value pairQ2053 |
| |  | | --- | | A.  HashMap |  |  | | --- | | B.  Hashtable |  |  | | --- | | C.  None | | |

|  |  |
| --- | --- |
| **363.** | Which class is used for sorting of every entry based on keyQ2054 |
| |  | | --- | | A.  HashMap |  |  | | --- | | B.  Hashtable |  |  | | --- | | C.  LinkedHashMap |  |  | | --- | | D.  TreeMap |  |  | | --- | | E.  None | | |
| **364.** | Map is anQ2055 |
| |  | | --- | | A.  Classs |  |  | | --- | | B.  interface |  |  | | --- | | C.  identifier |  |  | | --- | | D.  Keyword | | |

|  |  |
| --- | --- |
| **365.** | In which streams get() method availableQ2056 |
| |  | | --- | | A.  Queue |  |  | | --- | | B.  List |  |  | | --- | | C.  Map |  |  | | --- | | D.  both b and c |  |  | | --- | | E.  only c | | |
| **366.** | Is it possible to develop interface inside a another interfaceQ2057 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |

|  |  |
| --- | --- |
| **367.** | package iterators;  import java.util.ArrayList;  public class M1 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  }  }  Q2058 |
| |  | | --- | | A.  90  901  910  190  0  9 |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9] |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |
| **368.** | package iterators;  import java.util.ArrayList;  public class M2 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  for(int i = 0; i < list.size(); i++) {  System.out.println(list.get(i));  }  }  }  Q2059 |
| |  | | --- | | A.  90  901  910  190  0  9 |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9] |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **369.** | package iterators;  import java.util.ArrayList;  public class M3 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  for(Object element : list) {  System.out.print(element + ", ");  }  }  }  Q2060 |
| |  | | --- | | A.  90  901  910  190  0  9 |  |  | | --- | | B.  90,  901,  910,  190,  0,  9, |  |  | | --- | | C.  90, 901, 910, 190, 0, 9, | | |
| **370.** | package iterators;  import java.util.ArrayList;  import java.util.Iterator;  public class M4 {  public static void main(String[] args) {    ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  Iterator it = list.iterator();  while(it.hasNext()) {  System.out.print(it.next() + ", ");  }  }  }  Q2061 |
| |  | | --- | | A.  90  901  910  190  0  9 |  |  | | --- | | B.  90,  901,  910,  190,  0,  9, |  |  | | --- | | C.  90, 901, 910, 190, 0, 9, | | |

|  |  |
| --- | --- |
| **371.** | package iterators;  import java.util.ArrayList;  import java.util.ListIterator;  public class M5 {  public static void main(String[] args) {    ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  ListIterator it = list.listIterator();  while(it.hasNext()) {  System.out.print(it.next() + ", ");  }  }  }  Q2062 |
| |  | | --- | | A.  90  901  910  190  0  9 |  |  | | --- | | B.  90,  901,  910,  190,  0,  9, |  |  | | --- | | C.  90, 901, 910, 190, 0, 9, | | |
| **372.** | package iterators;  import java.util.ArrayList;  import java.util.Iterator;  public class M6 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  Iterator it = list.iterator();  while(it.hasNext()) {  System.out.print(it.next() + ", ");  it.remove();  }  System.out.println();  System.out.println(list);  }  }  Q2063 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  [] |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9, |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **373.** | package iterators;  import java.util.ArrayList;  import java.util.Iterator;  public class M7 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  Iterator it = list.iterator();  Object obj;  while(it.hasNext()) {  obj = it.next();  System.out.print(obj + ", ");  if(obj.equals(190)) {  it.remove();  }  }  System.out.println();  System.out.println(list);  }  }  Q2064 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  [90, 901, 910, 0, 9] |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9]  90,  901,  910,  190,  0,  9,  [90, 901, 910, 0, 9] |  |  | | --- | | C.  None | | |
| **374.** | package iterators;  import java.util.ArrayList;  import java.util.Iterator;  import java.util.function.Consumer;  public class M8 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  Iterator it = list.iterator();  Consumer consumer = new Consumer() {  public void accept(Object t) {  System.out.print(t + ", ");  }  };  it.forEachRemaining(consumer);  System.out.println();  System.out.println(list);  }  }  Q2065 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]  90,  901,  910,  190,  0,  9,  [90, 901, 910, 0, 9] |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  [90, 901, 910, 190, 0, 9] |  |  | | --- | | C.  None | | |

|  |  |
| --- | --- |
| **375.** | package iterators;  import java.util.ArrayList;  import java.util.Iterator;  import java.util.function.Consumer;  public class M9 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  Iterator it = list.iterator();  it.forEachRemaining(new Consumer() {  @Override  public void accept(Object t) {  System.out.print(t + ", ");  }  });  System.out.println();  System.out.println(list);  }  }  Q2066 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]  90,  901,  910,  190,  0,  9,  [90, 901, 910, 0, 9] |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  [90, 901, 910, 190, 0, 9] |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |
| **376.** | package iterators;  import java.util.ArrayList;  import java.util.Iterator;  import java.util.function.Consumer;  public class M10 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  Iterator it = list.iterator();  it.forEachRemaining((o1) -> System.out.println(o1));  System.out.println(list);  }  }  Q2067 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]  90,  901,  910,  190,  0,  9,  [90, 901, 910, 0, 9] |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  [90, 901, 910, 190, 0, 9] |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **377.** | package iterators;  import java.util.ArrayList;  import java.util.Iterator;  public class M11 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  Iterator it = list.iterator();  while(it.hasNext()) {  System.out.print(it.next() + ", ");  }  System.out.println();  System.out.println("-----------------------");  while(it.hasNext()) {  System.out.print(it.next() + ", ");  }  System.out.println(list);  }  }  Q2068 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  -----------------------  [90, 901, 910, 190, 0, 9] |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9]  90,  901,  910,  190,  0,  9,  -----------------------  [90, 901, 910, 190, 0, 9] |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |
| **378.** | package iterators;  import java.util.ArrayList;  import java.util.Iterator;  public class M12 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  Iterator it = list.iterator();  list.add(100);  while(it.hasNext()) {  System.out.print(it.next() + ", ");  }  System.out.println();  System.out.println(list);  }  }  Q2069 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  [90, 901, 910, 190, 0, 9, 100] |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9]  ConcurrentModificationException |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **379.** | package iterators;  import java.util.ArrayList;  import java.util.Iterator;  public class M13 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  Iterator it = list.iterator();    while(it.hasNext()) {  System.out.print(it.next() + ", ");  }  list.add(100);  System.out.println();  System.out.println(list);  }  }  Q2070 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  [90, 901, 910, 190, 0, 9, 100] |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9]  ConcurrentModificationException |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |
| **380.** | package iterators;  import java.util.ArrayList;  import java.util.Iterator;  import java.util.ListIterator;  public class M14 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  ListIterator it = list.listIterator();    while(it.hasNext()) {  System.out.print(it.next() + ", ");  }  System.out.println();  System.out.println("---------------");  while(it.hasPrevious()) {  System.out.print(it.previous() + ", ");  }  System.out.println();  System.out.println(list);  }  }  Q2071 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  ---------------  9, 0, 190, 910, 901, 90,  [90, 901, 910, 190, 0, 9] |  |  | | --- | | B.  Reverse reading is not possible |  |  | | --- | | C.  Compilation Error |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **381.** | package iterators;  import java.util.ArrayList;  import java.util.ListIterator;  public class M15 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  ListIterator it = list.listIterator();    while(it.hasNext()) {  System.out.print(it.next() + ", ");  }  System.out.println();  System.out.println("---------------");  while(it.hasPrevious()) {  System.out.print(it.previous() + ", ");  }  System.out.println();  System.out.println("---------------");  while(it.hasNext()) {  System.out.print(it.next() + ", ");  }  System.out.println();  System.out.println(list);  }  }  Q2072 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  ---------------  9, 0, 190, 910, 901, 90,  ---------------  [90, 901, 910, 190, 0, 9] |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  ---------------  ---------------  90, 901, 910, 190, 0, 9,  [90, 901, 910, 190, 0, 9] |  |  | | --- | | C.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  ---------------  9, 0, 190, 910, 901, 90,  ---------------  90, 901, 910, 190, 0, 9,  [90, 901, 910, 190, 0, 9] |  |  | | --- | | D.  None | | |
| **382.** | package iterators;  import java.util.ArrayList;  import java.util.ListIterator;  public class M18 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  ListIterator it = list.listIterator();  it.add(100);  while(it.hasNext()) {  System.out.print(it.next() + ", ");  }  System.out.println();  System.out.println(list);  }  }  Q2073 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  [100, 90, 901, 910, 190, 0, 9] |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  [90, 901, 910, 190, 0, 9, 100] |  |  | | --- | | C.  None | | |

|  |  |
| --- | --- |
| **383.** | package iterators;  import java.util.ArrayList;  import java.util.ListIterator;  public class M19 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  ListIterator it = list.listIterator();  Object obj;  while(it.hasNext()) {  obj = it.next();  if(obj.equals(910)) {  it.set(300);  }  System.out.print(it.next() + ", ");  }  System.out.println();  System.out.println(list);  }  }  Q2074 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]  910, 190, 0, 9,  [100, 90, 901, 910, 190, 0, 9] |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9]  901, 190, 9,  [90, 901, 300, 190, 0, 9] |  |  | | --- | | C.  [90, 901, 910, 190, 0, 9]  [90, 901, 300, 190, 0, 9] | | |
| **384.** | package iterators;  import java.util.ArrayList;  import java.util.ListIterator;  public class M19 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  ListIterator it = list.listIterator();  Object obj;  while(it.hasNext()) {  obj = it.next();  if(obj.equals(910)) {  it.set(300);  }  System.out.print(obj + ", ");  }  System.out.println();  System.out.println(list);  }  }  Q2075 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]    [90, 901, 910, 190, 0, 9] |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9]  901, 190, 9,  [90, 901, 300, 190, 0, 9] |  |  | | --- | | C.  [90, 901, 910, 190, 0, 9]  90, 901, 910, 190, 0, 9,  [90, 901, 300, 190, 0, 9] | | |

|  |  |
| --- | --- |
| **385.** | package iterators;  import java.util.ArrayList;  import java.util.ListIterator;  public class M20 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(90);  list.add(901);  list.add(910);  list.add(190);  list.add(0);  list.add(9);  System.out.println(list);  ListIterator it = list.listIterator();  Object obj;  while(it.hasNext()) {  System.out.print(it.previousIndex() + ":");  obj = it.next();  System.out.print(obj + ":");  System.out.print(it.nextIndex()+ ", ");  }  System.out.println();  System.out.println(list);  }  }  Q2076 |
| |  | | --- | | A.  [90, 901, 910, 190, 0, 9]  -1:90:1, 0:901:2, 1:910:3, 2:190:4, 3:0:5, 4:9:6,  [90, 901, 910, 190, 0, |  |  | | --- | | B.  [90, 901, 910, 190, 0, 9]  0:90:1, 1:901:2, 2:910:3, 3:190:4, 4:0:5, 5:9:6,  [90, 901, 910, 190, 0, | | |
| **386.** | package iterators;  import java.util.Vector;  public class M21 {  public static void main(String[] args) {  Vector v1 = new Vector();  v1.add(9000);  v1.add(100);  v1.add(9000000);  v1.add(0);  System.out.println(v1);  }  }  Q2077 |
| |  | | --- | | A.  [9000, 100, 9000000, 0] |  |  | | --- | | B.  [0, 100, 9000000, 9000] | | |

|  |  |
| --- | --- |
| **387.** | package iterators;  import java.util.Enumeration;  import java.util.Vector;  public class M22 {  public static void main(String[] args) {  Vector v1 = new Vector();  v1.add(9000);  v1.add(100);  v1.add(9000000);  v1.add(0);  Enumeration e1 = v1.elements();  while(e1.hasMoreElements()) {  System.out.println(e1.nextElement());  }  System.out.println("----------------------------");  while(e1.hasMoreElements()) {  System.out.println(e1.nextElement());  }  System.out.println("----------------------------");  System.out.println(v1);  }  }  Q2078 |
| |  | | --- | | A.  9000  100  9000000  0  ----------------------------  9000  100  9000000  0  ----------------------------  [9000, 100, 9000000, 0] |  |  | | --- | | B.  9000  100  9000000  0  ----------------------------  ----------------------------  [9000, 100, 9000000, 0] |  |  | | --- | | C.  ----------------------------  9000  100  9000000  0  ----------------------------  [9000, 100, 9000000, 0] | | |
| **388.** | package sorting;  import java.util.ArrayList;  import java.util.Collections;  public class M1 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(900);  list.add(901);  list.add(100);  list.add(990);  list.add(909);  list.add(400);  list.add(904);  System.out.println(list);  Collections.sort(list);  System.out.println(list);  }  }  Q2079 |
| |  | | --- | | A.  [900, 901, 100, 990, 909, 400, 904]  [100, 400, 900, 901, 904, 909, 990] |  |  | | --- | | B.  [900, 901, 100, 990, 909, 400, 904]  [900, 901, 100, 990, 909, 400, 904] |  |  | | --- | | C.  [100, 400, 900, 901, 904, 909, 990]  [100, 400, 900, 901, 904, 909, 990] |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **389.** | package sorting;  import java.util.ArrayList;  import java.util.Collections;  public class M2 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(900);  list.add(901);  list.add(100);  list.add(990);  list.add(909);  list.add(400);  list.add(904);  System.out.println(list);  Collections.sort(list, Collections.reverseOrder());  System.out.println(list);  }  }  Q2080 |
| |  | | --- | | A.  [900, 901, 100, 990, 909, 400, 904]  [100, 400, 900, 901, 904, 909, 990] |  |  | | --- | | B.  [900, 901, 100, 990, 909, 400, 904]  [900, 901, 100, 990, 909, 400, 904] |  |  | | --- | | C.  [900, 901, 100, 990, 909, 400, 904]  [990, 909, 904, 901, 900, 400, 100] | | |
| **390.** | package sorting;  import java.util.ArrayList;  import java.util.Collections;  public class M3 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add("xyz");  list.add("abc");  list.add("test");  list.add("java");  list.add("btm");  list.add("check");  System.out.println(list);  Collections.sort(list);  System.out.println(list);  }  }  Q2081 |
| |  | | --- | | A.  [xyz, abc, test, java, btm, check]  [abc, btm, check, java, test, xyz] |  |  | | --- | | B.  [xyz, abc, test, java, btm, check]  [xyz, abc, test, java, btm, check] |  |  | | --- | | C.  [abc, btm, check, java, test, xyz]  [abc, btm, check, java, test, xyz] | | |

|  |  |
| --- | --- |
| **391.** | package sorting;  import java.util.ArrayList;  import java.util.Collections;  public class M4 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add("xyz");  list.add("abc");  list.add("test");  list.add("java");  list.add("btm");  list.add("check");  System.out.println(list);  Collections.sort(list, Collections.reverseOrder());  System.out.println(list);  }  }  Q2082 |
| |  | | --- | | A.  [xyz, abc, test, java, btm, check]  [abc, btm, check, java, test, xyz] |  |  | | --- | | B.  [xyz, abc, test, java, btm, check]  [xyz, abc, test, java, btm, check] |  |  | | --- | | C.  [xyz, abc, test, java, btm, check]  [xyz, test, java, check, btm, abc] | | |
| **392.** | package sorting;  import java.util.ArrayList;  import java.util.Collections;  public class M5{  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add("xyz");  list.add("abc");  list.add("test");  list.add("java");  list.add("btm");  list.add("check");  list.add("6787");  list.add("98778");  System.out.println(list);  Collections.sort(list);  System.out.println(list);  }  }  Q2083 |
| |  | | --- | | A.  Compilation Error |  |  | | --- | | B.  [xyz, abc, test, java, btm, check, 6787, 98778]  [6787, 98778, abc, btm, check, java, test, xyz] |  |  | | --- | | C.  [xyz, abc, test, java, btm, check, 6787, 98778]  [xyz, abc, test, java, btm, check, 6787, 98778] | | |

|  |  |
| --- | --- |
| **393.** | package sorting;  import java.util.ArrayList;  import java.util.Collections;  public class M6{  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(900);  list.add(901);  list.add(100);  list.add(990);  list.add(909.0);  list.add(400);  list.add(904);  System.out.println(list);  Collections.sort(list);  System.out.println(list);  }  }  Q2084 |
| |  | | --- | | A.  [900, 901, 100, 990, 909, 400, 904]  [100, 400, 900, 901, 904, 909, 990] |  |  | | --- | | B.  [900, 901, 100, 990, 909.0, 400, 904]  ClassCastException |  |  | | --- | | C.  Compilation Error | | |
| **394.** | package sorting;  import java.util.ArrayList;  import java.util.Collections;  public class M7 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(900);  list.add(901);  list.add(100);  list.add(990);  list.add(1);  list.add(400);  list.add(904);  System.out.println(list);  Collections.sort(list);  System.out.println(list);  }  }  Q2085 |
| |  | | --- | | A.  [900, 901, 100, 990, null, 400, 904]  [null, 100, 400, 900, 901, 904, 990] |  |  | | --- | | B.  [900, 901, 100, 990, null, 400, 904]  NullPointerException |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **395.** | package sorting;  import java.util.ArrayList;  import java.util.Collections;  class A{  int i;  A(int i){  this.i= i;  }  @Override  public String toString() {  // TODO Auto-generated method stub  return "i = " + i;  }  }  public class M8 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(new A(90));  list.add(new A(9));  list.add(new A(0));  list.add(new A(80));  System.out.println(list);  Collections.sort(list);  System.out.println(list);  }  }  Q2086 |
| |  | | --- | | A.  [i = 90, i = 9, i = 0, i = 80]  [i = 0, i = 9, i = 80, i = 90] |  |  | | --- | | B.  [i = 90, i = 9, i = 0, i = 80]  ClassCastException |  |  | | --- | | C.  Compilation Error | | |
| **396.** | Single method interface is also called as functional interfaceQ2087 |
| |  | | --- | | A.  True |  |  | | --- | | B.  False | | |

|  |  |
| --- | --- |
| **397.** | package sorting;  import java.util.ArrayList;  import java.util.Collections;  class B implements Comparable {  int i;  B(int i){  this.i= i;  }  @Override  public String toString() {  // TODO Auto-generated method stub  return "i = " + i;  }  @Override  public int compareTo(Object o) {  return i - ((B)o).i;  }  }  public class M9 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(new B(90));  list.add(new B(9));  list.add(new B(0));  list.add(new B(80));  System.out.println(list);  Collections.sort(list);  System.out.println(list);  }  }  Q2088 |
| |  | | --- | | A.  [i = 90, i = 9, i = 0, i = 80]  [i = 0, i = 9, i = 80, i = 90] |  |  | | --- | | B.  [i = 90, i = 9, i = 0, i = 80]  ClassCastException |  |  | | --- | | C.  Compilation Error | | |
| **398.** | package sorting;  import java.util.ArrayList;  import java.util.Collections;  import java.util.Comparator;  class C{  int i, j;  public C(int i, int j) {  super();  this.i = i;  this.j = j;  }  @Override  public String toString() {  return "C [i=" + i + ", j=" + j + "]";  }  }  class SortBasedOnIValue implements Comparator{  @Override  public int compare(Object o1, Object o2) {  return ((C)o1).i - ((C)o2).i;  }  }  class SortBasedOnJValue implements Comparator{  @Override  public int compare(Object o1, Object o2) {  return ((C)o1).j - ((C)o2).j;  }  }  public class M10 {  public static void main(String[] args) {  ArrayList list = new ArrayList();  list.add(new C(10, 20));  list.add(new C(20, 10));  list.add(new C(5, 15));  list.add(new C(15, 8));  System.out.println(list);  Collections.sort(list, new SortBasedOnIValue());  System.out.println(list);  Collections.sort(list, new SortBasedOnJValue());  System.out.println(list);  }  }  Q2089 |
| |  | | --- | | A.  [C [i=10, j=20], C [i=20, j=10], C [i=5, j=15], C [i=15, j=8]]  [C [i=5, j=15], C [i=10, j=20], C [i=15, j=8], C [i=20, j=10]]  [C [i=15, j=8], C [i=20, j=10], C [i=5, j=15], C [i=10, j=20]] |  |  | | --- | | A.  [C [i=10, j=20], C [i=20, j=10], C [i=5, j=15], C [i=15, j=8]]  [C [i=15, j=8], C [i=20, j=10], C [i=5, j=15], C [i=10, j=20]]  [C [i=5, j=15], C [i=10, j=20], C [i=15, j=8], C [i=20, j=10]] | | |

|  |  |
| --- | --- |
| **399.** | package mapBasedSorting;  import java.util.HashMap;  import java.util.Set;  public class M1 {  public static void main(String[] args) {  HashMap map = new HashMap();  map.put("hello", 201);  map.put("abc", 201);  map.put("xyz", 201);  map.put("test", 201);  map.put("btm", 201);  map.put("java", 201);    Set entries = map.entrySet();    for(Object obj : entries) {  System.out.println(obj);  }  }  }  //How the output will beQ2090 |
| |  | | --- | | A.  same order |  |  | | --- | | B.  in order output |  |  | | --- | | C.  None | | |
| **400.** | package mapBasedSorting;  import java.util.HashMap;  import java.util.Map.Entry;  import java.util.Set;  public class M2 {  public static void main(String[] args) {  HashMap map = new HashMap();  map.put("hello", 201);  map.put("abc", 20);  map.put("xyz", 21);  map.put("test", 1);  map.put("btm", 25);  map.put("java", 101);    Set entries = map.entrySet();  Entry entry;  for(Object obj : entries) {  entry = (Entry) obj;  System.out.println(entry.getKey() + ":" + entry.getValue());  }  }  }  //How the output will beQ2091 |
| |  | | --- | | A.  same order |  |  | | --- | | B.  in order output |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **401.** | package mapBasedSorting;  import java.util.HashMap;  import java.util.TreeMap;  public class M3 {  public static void main(String[] args) {  HashMap map = new HashMap();  map.put("hello", 201);  map.put("abc", 20);  map.put("xyz", 21);  map.put("test", 1);  map.put("btm", 25);  map.put("java", 101);  TreeMap map1 = new TreeMap();  map1.putAll(map);  System.out.println(map1);  }  }  Q2092 |
| |  | | --- | | A.  {abc=20, btm=25, hello=201, java=101, test=1, xyz=21} |  |  | | --- | | B.  {abc:20, java:101, test:1, xyz:21, btm:25, hello:201} |  |  | | --- | | C.  {  abc=20,  btm=25,  hello=201,  java=101,  test=1,  xyz=21  } | | |
| **402.** | package mapBasedSorting;  import java.util.Comparator;  import java.util.HashMap;  import java.util.Map.Entry;  import java.util.Set;  import java.util.TreeSet;  public class M4 {  public static void main(String[] args) {  HashMap map = new HashMap();  map.put("hello", 201);  map.put("abc", 20);  map.put("xyz", 21);  map.put("test", 1);  map.put("btm", 25);  map.put("java", 101);    Set entries = map.entrySet();    Comparator c1 = new Comparator() {  public int compare(Object o1, Object o2) {  Entry e1 = (Entry) o1;  Entry e2 = (Entry) o2;  return ((Integer)e1.getValue()).compareTo((Integer)e2.getValue()) ;  }  };  TreeSet set = new TreeSet(c1);  set.addAll(entries);  System.out.println(set);  }  }  Q2093 |
| |  | | --- | | A.  [test=1, abc=20, xyz=21, btm=25, java=101, hello=201] |  |  | | --- | | B.  {abc=20, btm=25, hello=201, java=101, test=1, xyz=21} |  |  | | --- | | C.  Compilation Error | | |

|  |  |
| --- | --- |
| **403.** | Which are synchronized by defaultQ2094 |
| |  | | --- | | A.  vector |  |  | | --- | | B.  Hashtable |  |  | | --- | | C.  Hashmap |  |  | | --- | | D.  Both a and b |  |  | | --- | | E.  only c | | |
| **404.** | In vector object how many threads are allowed to enter at a timeQ2095 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  2 |  |  | | --- | | C.  3 |  |  | | --- | | D.  Multiple threads | | |

|  |  |
| --- | --- |
| **405.** | In Hashtable object how many threads are allowed to enter at a timeQ2096 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  2 |  |  | | --- | | C.  3 |  |  | | --- | | D.  Multiple threads | | |
| **406.** | Which of the fillowing objects are not possible to use in multi threaded environment stright awayQ2097 |
| |  | | --- | | A.  ArrayList |  |  | | --- | | B.  Vector |  |  | | --- | | C.  Hashtable |  |  | | --- | | D.  None | | |

|  |  |
| --- | --- |
| **407.** | Which of the following threads allows only one thread at a time Q2098 |
| |  | | --- | | A.  Non synchronized thread |  |  | | --- | | B.  Synchronized thread |  |  | | --- | | C.  None | | |
| **408.** | One object can be shared to how many threadsQ2099 |
| |  | | --- | | A.  1 |  |  | | --- | | B.  2 |  |  | | --- | | C.  3 |  |  | | --- | | D.  any number of threads |  |  | | --- | | E.  None | | |

|  |  |
| --- | --- |
| **409.** | Is it possible to make ArrayList as synchronizedQ2100 |
| |  | | --- | | A.  Yes |  |  | | --- | | B.  No | | |
| **410.** | package collectionsSynchronized;  import java.util.List;  import java.util.ArrayList;  import java.util.Collections;  public class M1 {  public static void main(String[] args) {  List list = new ArrayList();  list.add(90);  list.add(90);  list.add(90);  list.add(90);  System.out.println(list);    list = Collections.synchronizedList(list);    }  }  Q2101 |
| |  | | --- | | A.  [90, 90, 90, 90] |  |  | | --- | | B.  Compilation Error |  |  | | --- | | C.  None | | |

|  |  |
| --- | --- |
| **411.** | package collectionsSynchronized;  import java.util.Collections;  import java.util.HashSet;  import java.util.Set;  public class M2 {  public static void main(String[] args) {  Set set = new HashSet();// line 9    set = Collections.synchronizedSet(set);// line 13  }  }  // Check this program after line 13 it is thread safe are notQ2102 |
| |  | | --- | | A.  Thread safe |  |  | | --- | | B.  Not safe | | |
| **412.** | package collectionsSynchronized;  import java.util.Collections;  import java.util.HashMap;  import java.util.Map;  public class M3 {  public static void main(String[] args) {  Map map = new HashMap(); // line 9    map = Collections.synchronizedMap(map); // line 11  }  }  // Check this program after line 9 it is thread safe are notQ2103 |
| |  | | --- | | A.  Thread safe |  |  | | --- | | B.  Not safe | | |

|  |  |
| --- | --- |
| **413.** | can we define an empty array?Q2825 |
| |  | | --- | | A.  yes |  |  | | --- | | B.  no | | |
| **414.** | array size must be declared in which data type?Q2826 |
| |  | | --- | | A.  boolean |  |  | | --- | | B.  double |  |  | | --- | | C.  null |  |  | | --- | | D.  int | | |

|  |  |
| --- | --- |
| **415.** | array index must be declared in which data type?Q2827 |
| |  | | --- | | A.  boolean |  |  | | --- | | B.  double |  |  | | --- | | C.  null |  |  | | --- | | D.  int | | |
| **416.** | arrays are which datatype in case of java?Q2828 |
| |  | | --- | | A.  derived datatype |  |  | | --- | | B.  primitive datatype | | |

|  |  |
| --- | --- |
| **417.** | to read the elements of an arraylist which method was used?Q2829 |
| |  | | --- | | A.  set() |  |  | | --- | | B.  get() |  |  | | --- | | C.  size() |  |  | | --- | | D.  read() | | |
| **418.** | Injava arrays are what?Q2830 |
| |  | | --- | | A.  object |  |  | | --- | | B.  object references |  |  | | --- | | C.  primitive data type |  |  | | --- | | D.  none of the above | | |

|  |  |
| --- | --- |
| **419.** | which one is a valid statement?Q2831 |
| |  | | --- | | A.  char[] c=new char(); |  |  | | --- | | B.  char[] c=new char[5]; |  |  | | --- | | C.  char[] c=new char(4); |  |  | | --- | | D.  char[] c=new char[]; | | |
| **420.** | public class Test{  public static void main(String[] args){  int[] a=new int[0];  System.out.print(a.length);}}  what is the output?Q2832 |
| |  | | --- | | A.  0 |  |  | | --- | | B.  compile time error as arrays can not initialized as 0 |  |  | | --- | | C.  compile time error as it is a.length() not a.length |  |  | | --- | | D.  none of the above | | |

|  |  |
| --- | --- |
| **421.** | public class Test{  public static void main(String[] args){  int[] x=new int[3];  System.out.println("x[0] is"+x[0]);}}  what is the output?Q2833 |
| |  | | --- | | A.  compile time error |  |  | | --- | | B.  run time error |  |  | | --- | | C.  x[0] is 0 |  |  | | --- | | D.  none of the above | | |
| **422.** | int[] arr=new int[5];  arr=new int[6];  what is the output?Q2834 |
| |  | | --- | | A.  compile time error |  |  | | --- | | B.  run time error |  |  | | --- | | C.  no error |  |  | | --- | | D.  none of the above | | |