# Java Programming Test 10

[Home](http://www.indiabix.com/) » [Online Test](http://www.indiabix.com/online-test/categories/) » [Online Java Programming Test](http://www.indiabix.com/online-test/java-programming-test/) » Java Programming Test 10

|  |  |  |
| --- | --- | --- |
| **Marks : 0/20** | | |
| Total number of questions | : | **20** |
| Number of answered questions | : | **0** |
| Number of unanswered questions | : | **20** |

### Test Review : View answers and explanation for this test.

|  |  |
| --- | --- |
| 1. | What will be the output of the program?  public class Test  {  public int aMethod()  {  static int i = 0;  i++;  return i;  }  public static void main(String args[])  {  Test test = new Test();  test.aMethod();  int j = test.aMethod();  System.out.println(j);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 0 | | |  | B. | |  | | --- | | 1 | | |  | C. | |  | | --- | | 2 | | |  | D. | |  | | --- | | Compilation fails. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Compilation failed because static was an illegal start of expression - method variables do not have a modifier (they are always considered local).  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-45) |

|  |  |
| --- | --- |
| 2. | What will be the output of the program?  class PassA  {  public static void main(String [] args)  {  PassA p = new PassA();  p.start();  }  void start()  {  long [] a1 = {3,4,5};  long [] a2 = fix(a1);  System.out.print(a1[0] + a1[1] + a1[2] + " ");  System.out.println(a2[0] + a2[1] + a2[2]);  }  long [] fix(long [] a3)  {  a3[1] = 7;  return a3;  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 12 15 | | |  | B. | |  | | --- | | 15 15 | | |  | C. | |  | | --- | | 3 4 5 3 7 5 | | |  | D. | |  | | --- | | 3 7 5 3 7 5 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Output: 15 15  The reference variables a1 and a3 refer to the same long array object. When the [1] element is updated in the fix() method, it is updating the array referred to by a1. The reference variable a2 refers to the same array object.  So Output: 3+7+5+" "3+7+5  Output: 15 15 Because Numeric values will be added  Learn more problems on : [Operators and Assignments](http://www.indiabix.com/java-programming/operators-and-assignments/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/operators-and-assignments/discussion-70) |

|  |  |
| --- | --- |
| 3. | What will be the output of the program?  class Equals  {  public static void main(String [] args)  {  int x = 100;  double y = 100.1;  boolean b = (x = y); /\* Line 7 \*/  System.out.println(b);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | true | | |  | B. | |  | | --- | | false | | |  | C. | |  | | --- | | Compilation fails | | |  | D. | |  | | --- | | An exception is thrown at runtime | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  The code will not compile because in line 7, the line will work only if we use (x==y) in the line. The ==operator compares values to produce a boolean, whereas the = operator assigns a value to variables.  Option A, B, and D are incorrect because the code does not get as far as compiling. If we corrected this code, the output would be false.  Learn more problems on : [Operators and Assignments](http://www.indiabix.com/java-programming/operators-and-assignments/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/operators-and-assignments/discussion-58) |

|  |  |
| --- | --- |
| 4. | What will be the output of the program?  class Bitwise  {  public static void main(String [] args)  {  int x = 11 & 9;  int y = x ^ 3;  System.out.println( y | 12 );  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 0 | | |  | B. | |  | | --- | | 7 | | |  | C. | |  | | --- | | 8 | | |  | D. | |  | | --- | | 14 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  The & operator produces a 1 bit when both bits are 1. The result of the & operation is 9. The ^ operator produces a 1 bit when exactly one bit is 1; the result of this operation is 10. The | operator produces a 1 bit when at least one bit is 1; the result of this operation is 14.  Learn more problems on : [Operators and Assignments](http://www.indiabix.com/java-programming/operators-and-assignments/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/operators-and-assignments/discussion-60) |

|  |  |
| --- | --- |
| 5. | What will be the output of the program?  class SSBool  {  public static void main(String [] args)  {  boolean b1 = true;  boolean b2 = false;  boolean b3 = true;  if ( b1 & b2 | b2 & b3 | b2 ) /\* Line 8 \*/  System.out.print("ok ");  if ( b1 & b2 | b2 & b3 | b2 | b1 ) /\*Line 10\*/  System.out.println("dokey");  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | ok | | |  | B. | |  | | --- | | dokey | | |  | C. | |  | | --- | | ok dokey | | |  | D. | |  | | --- | | No output is produced | | |  | E. | |  | | --- | | Compilation error | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  The & operator has a higher precedence than the | operator so that on line 8 b1 and b2 are evaluated together as are b2 & b3. The final b1 in line 10 is what causes that if test to be true. Hence it prints "dokey".  Learn more problems on : [Operators and Assignments](http://www.indiabix.com/java-programming/operators-and-assignments/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/operators-and-assignments/discussion-63) |

|  |  |
| --- | --- |
| 6. | What will be the output of the program?  int x = l, y = 6;  while (y--)  {  x++;  }  System.out.println("x = " + x +" y = " + y); |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | x = 6 y = 0 | | |  | B. | |  | | --- | | x = 7 y = 0 | | |  | C. | |  | | --- | | x = 6 y = -1 | | |  | D. | |  | | --- | | Compilation fails. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Compilation fails because the while loop demands a boolean argument for it's looping condition, but in the code, it's given an int argument.  while(true) { //insert code here }  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-100) |

|  |  |
| --- | --- |
| 7. | What will be the output of the program?  for (int i = 0; i < 4; i += 2)  {  System.out.print(i + " ");  }  System.out.println(i); /\* Line 5 \*/ |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 0 2 4 | | |  | B. | |  | | --- | | 0 2 4 5 | | |  | C. | |  | | --- | | 0 1 2 3 4 | | |  | D. | |  | | --- | | Compilation fails. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Compilation fails on the line 5 - System.out.println(i); as the variable i has only been declared within the for loop. It is not a recognised variable outside the code block of loop.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-102) |

|  |  |
| --- | --- |
| 8. | What will be the output of the program?  int x = 3;  int y = 1;  if (x = y) /\* Line 3 \*/  {  System.out.println("x =" + x);  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | x = 1 | | |  | B. | |  | | --- | | x = 3 | | |  | C. | |  | | --- | | Compilation fails. | | |  | D. | |  | | --- | | The code runs with no output. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Line 3 uses an assignment as opposed to comparison. Because of this, the if statement receives an integer value instead of a boolean. And so the compilation fails.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-91) |

|  |  |
| --- | --- |
| 9. | What will be the output of the program?  public class X  {  public static void main(String [] args)  {  try  {  badMethod();  System.out.print("A");  }  catch (RuntimeException ex) /\* Line 10 \*/  {  System.out.print("B");  }  catch (Exception ex1)  {  System.out.print("C");  }  finally  {  System.out.print("D");  }  System.out.print("E");  }  public static void badMethod()  {  throw new RuntimeException();  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | BD | | |  | B. | |  | | --- | | BCD | | |  | C. | |  | | --- | | BDE | | |  | D. | |  | | --- | | BCDE | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  A Run time exception is thrown and caught in the catch statement on line 10. All the code after the finally statement is run because the exception has been caught.  Learn more problems on : [Exceptions](http://www.indiabix.com/java-programming/exceptions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/exceptions/discussion-112) |

|  |  |
| --- | --- |
| 10. | public class ExceptionTest  {  class TestException extends Exception {}  public void runTest() throws TestException {}  public void test() /\* Point X \*/  {  runTest();  }  }  At Point X on line 5, which code is necessary to make the code compile? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | No code is necessary. | | |  | B. | |  | | --- | | throws Exception | | |  | C. | |  | | --- | | catch ( Exception e ) | | |  | D. | |  | | --- | | throws RuntimeException | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B is correct. This works because it DOES throw an exception if an error occurs.  Option A is wrong. If you compile the code as given the compiler will complain:  "unreported exception must be caught or declared to be thrown" The class extends Exception so we are forced to test for exceptions.  Option C is wrong. The catch statement belongs in a method body not a method specification.  Option D is wrong. TestException is a subclass of Exception therefore the test method, in this example, must throw TestException or some other class further up the Exception tree. Throwing RuntimeException is just not on as this belongs in the java.lang.RuntimeException branch (it is not a superclass of TestException). The compiler complains with the same error as in A above.  Learn more problems on : [Exceptions](http://www.indiabix.com/java-programming/exceptions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/exceptions/discussion-117) |

|  |  |
| --- | --- |
| 11. | You need to store elements in a collection that guarantees that no duplicates are stored. Which one of the following interfaces provide that capability? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Java.util.Map | | |  | B. | |  | | --- | | Java.util.List | | |  | C. | |  | | --- | | Java.util.Collection | | |  | D. | |  | | --- | | None of the above | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Option A is correct. A Map cannot contain duplicate keys.  Option B is wrong. Lists typically allow duplicate elements.  Option C is wrong. Collection allows duplicate elements.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-130) |

|  |  |
| --- | --- |
| 12. | Which collection class allows you to access its elements by associating a key with an element's value, and provides synchronization? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | java.util.SortedMap | | |  | B. | |  | | --- | | java.util.TreeMap | | |  | C. | |  | | --- | | java.util.TreeSet | | |  | D. | |  | | --- | | java.util.Hashtable | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Hashtable is the only class listed that provides synchronized methods. If you need synchronization great; otherwise, use HashMap, it's faster.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-123) |

|  |  |
| --- | --- |
| 13. | What is the numerical range of char? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 0 to 32767 | | |  | B. | |  | | --- | | 0 to 65535 | | |  | C. | |  | | --- | | -256 to 255 | | |  | D. | |  | | --- | | -32768 to 32767 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  The char type is integral but unsigned. The range of a variable of type char is from 0 to 216-1 or 0 to 65535. Java characters are Unicode, which is a 16-bit encoding capable of representing a wide range of international characters. If the most significant nine bits of a char are 0, then the encoding is the same as seven-bit ASCII.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-127) |

|  |  |
| --- | --- |
| 14. | Which of the following are true statements?   1. The Iterator interface declares only three methods: hasNext, next and remove. 2. The ListIterator interface extends both the List and Iterator interfaces. 3. The ListIterator interface provides forward and backward iteration capabilities. 4. The ListIterator interface provides the ability to modify the List during iteration. 5. The ListIterator interface provides the ability to determine its position in the List. |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 2, 3, 4 and 5 | | |  | B. | |  | | --- | | 1, 3, 4 and 5 | | |  | C. | |  | | --- | | 3, 4 and 5 | | |  | D. | |  | | --- | | 1, 2 and 3 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  The ListIterator interface extends the Iterator interface and declares additional methods to provide forward and backward iteration capabilities, List modification capabilities, and the ability to determine the position of the iterator in the List.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-148) |

|  |  |
| --- | --- |
| 15. | Which method must be defined by a class implementing the java.lang.Runnable interface? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | void run() | | |  | B. | |  | | --- | | public void run() | | |  | C. | |  | | --- | | public void start() | | |  | D. | |  | | --- | | void run(int priority) | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B is correct because in an interface all methods are abstract by default therefore they must be overridden by the implementing class. The Runnable interface only contains 1 method, the void run()method therefore it must be implemented.  Option A and D are incorrect because they are narrowing the access privileges i.e. package(default) access is narrower than public access.  Option C is not method in the Runnable interface therefore it is incorrect.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-169) |

|  |  |
| --- | --- |
| 16. | Which two statements are true?   1. Deadlock will not occur if wait()/notify() is used 2. A thread will resume execution as soon as its sleep duration expires. 3. Synchronization can prevent two objects from being accessed by the same thread. 4. The wait() method is overloaded to accept a duration. 5. The notify() method is overloaded to accept a duration. 6. Both wait() and notify() must be called from a synchronized context. |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 2 | | |  | B. | |  | | --- | | 3 and 5 | | |  | C. | |  | | --- | | 4 and 6 | | |  | D. | |  | | --- | | 1 and 3 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Statements (4) and (6) are correct. (4) is correct because the wait() method is overloaded to accept a wait duration in milliseconds. If the thread has not been notified by the time the wait duration has elapsed, then the thread will move back to runnable even without having been notified.  (6) is correct because wait()/notify()/notifyAll() must all be called from within a synchronized, context. A thread must own the lock on the object its invoking wait()/notify()/notifyAll() on.  (1) is incorrect because wait()/notify() will not prevent deadlock.  (2) is incorrect because a sleeping thread will return to runnable when it wakes up, but it might not necessarily resume execution right away. To resume executing, the newly awakened thread must still be moved from runnable to running by the scheduler.  (3) is incorrect because synchronization prevents two or more threads from accessing the same object.  (5) is incorrect because notify() is not overloaded to accept a duration.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-199) |

|  |  |
| --- | --- |
| 17. | Which of the following statements is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | If assertions are compiled into a source file, and if no flags are included at runtime, assertions will execute by default. | | |  | B. | |  | | --- | | As of Java version 1.4, assertion statements are compiled by default. | | |  | C. | |  | | --- | | With the proper use of runtime arguments, it is possible to instruct the VM to disable assertions for a certain class, and to enable assertions for a certain package, at the same time. | | |  | D. | |  | | --- | | When evaluating command-line arguments, the VM gives -ea flags precedence over -da flags. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Option C is true because multiple VM flags can be used on a single invocation of a Java program.  Option A is incorrect because at runtime assertions are ignored by default.  Option B is incorrect because as of Java 1.4 you must add the argument -source 1.4 to the command line if you want the compiler to compile assertion statements.  Option D is incorrect because the VM evaluates all assertion flags left to right.  Learn more problems on : [Assertions](http://www.indiabix.com/java-programming/assertions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/assertions/discussion-221) |

|  |  |
| --- | --- |
| 18. | What will be the output of the program?  try  {  Float f1 = new Float("3.0");  int x = f1.intValue();  byte b = f1.byteValue();  double d = f1.doubleValue();  System.out.println(x + b + d);  }  catch (NumberFormatException e) /\* Line 9 \*/  {  System.out.println("bad number"); /\* Line 11 \*/  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 9.0 | | |  | B. | |  | | --- | | bad number | | |  | C. | |  | | --- | | Compilation fails on line 9. | | |  | D. | |  | | --- | | Compilation fails on line 11. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  The xxxValue() methods convert any numeric wrapper object's value to any primitive type. When narrowing is necessary, significant bits are dropped and the results are difficult to calculate.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-240) |

|  |  |
| --- | --- |
| 19. | What will be the output of the program?  public class ExamQuestion6  {  static int x;  boolean catch()  {  x++;  return true;  }  public static void main(String[] args)  {  x=0;  if ((catch() | catch()) || catch())  x++;  System.out.println(x);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 | | |  | B. | |  | | --- | | 2 | | |  | C. | |  | | --- | | 3 | | |  | D. | |  | | --- | | Compilation Fails | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Initially this looks like a question about the logical and logical shortcut operators "|" and "||" but on closer inspection it should be noticed that the name of the boolean method in this code is "catch". "catch" is a reserved keyword in the Java language and cannot be used as a method name. Hence Compilation will fail.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-252) |

|  |  |
| --- | --- |
| 20. | What will be the output of the program?  String s = "hello";  Object o = s;  if( o.equals(s) )  {  System.out.println("A");  }  else  {  System.out.println("B");  }  if( s.equals(o) )  {  System.out.println("C");  }  else  {  System.out.println("D");  }   1. A 2. B 3. C 4. D |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 3 | | |  | B. | |  | | --- | | 2 and 4 | | |  | C. | |  | | --- | | 3 and 4 | | |  | D. | |  | | --- | | 1 and 2 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-255) |

# Java Programming Test 9

[Home](http://www.indiabix.com/) » [Online Test](http://www.indiabix.com/online-test/categories/) » [Online Java Programming Test](http://www.indiabix.com/online-test/java-programming-test/) » Java Programming Test 9

|  |  |  |
| --- | --- | --- |
| **Marks : 0/20** | | |
| Total number of questions | : | **20** |
| Number of answered questions | : | **0** |
| Number of unanswered questions | : | **20** |

### Test Review : View answers and explanation for this test.

|  |  |
| --- | --- |
| 1. | What will be the output of the program?  class Super  {  public int i = 0;  public Super(String text) /\* Line 4 \*/  {  i = 1;  }  }  class Sub extends Super  {  public Sub(String text)  {  i = 2;  }  public static void main(String args[])  {  Sub sub = new Sub("Hello");  System.out.println(sub.i);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 0 | | |  | B. | |  | | --- | | 1 | | |  | C. | |  | | --- | | 2 | | |  | D. | |  | | --- | | Compilation fails. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  A default no-args constructor is not created because there is a constructor supplied that has an argument, line 4. Therefore the sub-class constructor must explicitly make a call to the super class constructor:  public Sub(String text)  {  super(text); // this must be the first line constructor  i = 2;  }  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-46) |

|  |  |
| --- | --- |
| 2. | What will be the output of the program?  class Base  {  Base()  {  System.out.print("Base");  }  }  public class Alpha extends Base  {  public static void main(String[] args)  {  new Alpha(); /\* Line 12 \*/  new Base(); /\* Line 13 \*/  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Base | | |  | B. | |  | | --- | | BaseBase | | |  | C. | |  | | --- | | Compilation fails | | |  | D. | |  | | --- | | The code runs with no output | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B is correct. It would be correct if the code had compiled, and the subclass Alpha had been saved in its own file. In this case Java supplies an implicit call from the sub-class constructor to the no-args constructor of the super-class therefore line 12 causes Base to be output. Line 13 also causes Base to be output.  Option A is wrong. It would be correct if either the main class or the subclass had not been instantiated.  Option C is wrong. The code compiles.  Option D is wrong. There is output.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-47) |

|  |  |
| --- | --- |
| 3. | What will be the output of the program?  import java.util.\*;  public class NewTreeSet2 extends NewTreeSet  {  public static void main(String [] args)  {  NewTreeSet2 t = new NewTreeSet2();  t.count();  }  }  protected class NewTreeSet  {  void count()  {  for (int x = 0; x < 7; x++,x++ )  {  System.out.print(" " + x);  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 0 2 4 | | |  | B. | |  | | --- | | 0 2 4 6 | | |  | C. | |  | | --- | | Compilation fails at line 2 | | |  | D. | |  | | --- | | Compilation fails at line 10 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Nonnested classes cannot be marked protected (or final for that matter), so the compiler will fail at protected class NewTreeSet.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-43) |

|  |  |
| --- | --- |
| 4. | Which two statements are true for any concrete class implementing the java.lang.Runnable interface?   1. You can extend the Runnable interface as long as you override the public run() method. 2. The class must contain a method called run() from which all code for that thread will be initiated. 3. The class must contain an empty public void method named run(). 4. The class must contain a public void method named runnable(). 5. The class definition must include the words implements Threads and contain a method called run(). 6. The mandatory method must be public, with a return type of void, must be called run(), and cannot take any arguments. |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 3 | | |  | B. | |  | | --- | | 2 and 4 | | |  | C. | |  | | --- | | 1 and 5 | | |  | D. | |  | | --- | | 2 and 6 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  (2) and (6). When a thread's run() method completes, the thread will die. The run() method must be declared public void and not take any arguments.  (1) is incorrect because classes can never extend interfaces. (3) is incorrect because the run() method is typically not empty; if it were, the thread would do nothing. (4) is incorrect because the mandatory method is run(). (5) is incorrect because the class implements Runnable.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-52) |

|  |  |
| --- | --- |
| 5. | What will be the output of the program?  public class X  {  public static void main(String [] args)  {  try  {  badMethod();  System.out.print("A");  }  catch (Exception ex)  {  System.out.print("B");  }  finally  {  System.out.print("C");  }  System.out.print("D");  }  public static void badMethod()  {  throw new Error(); /\* Line 22 \*/  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | ABCD | | |  | B. | |  | | --- | | Compilation fails. | | |  | C. | |  | | --- | | C is printed before exiting with an error message. | | |  | D. | |  | | --- | | BC is printed before exiting with an error message. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Error is thrown but not recognised line(22) because the only catch attempts to catch an Exception and Exception is not a superclass of Error. Therefore only the code in the finally statement can be run before exiting with a runtime error (Exception in thread "main" java.lang.Error).  Learn more problems on : [Exceptions](http://www.indiabix.com/java-programming/exceptions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/exceptions/discussion-110) |

|  |  |
| --- | --- |
| 6. | What will be the output of the program?  public class X  {  public static void main(String [] args)  {  try  {  badMethod(); /\* Line 7 \*/  System.out.print("A");  }  catch (Exception ex) /\* Line 10 \*/  {  System.out.print("B"); /\* Line 12 \*/  }  finally /\* Line 14 \*/  {  System.out.print("C"); /\* Line 16 \*/  }  System.out.print("D"); /\* Line 18 \*/  }  public static void badMethod()  {  throw new RuntimeException();  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | AB | | |  | B. | |  | | --- | | BC | | |  | C. | |  | | --- | | ABC | | |  | D. | |  | | --- | | BCD | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  (1) A RuntimeException is thrown, this is a subclass of exception.  (2) The exception causes the try to complete abruptly (line 7) therefore line 8 is never executed.  (3) The exception is caught (line 10) and "B" is output (line 12)  (4) The finally block (line 14) is always executed and "C" is output (line 16).  (5) The exception was caught, so the program continues with line 18 and outputs "D".  Learn more problems on : [Exceptions](http://www.indiabix.com/java-programming/exceptions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/exceptions/discussion-109) |

|  |  |
| --- | --- |
| 7. | Which interface does java.util.Hashtable implement? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Java.util.Map | | |  | B. | |  | | --- | | Java.util.List | | |  | C. | |  | | --- | | Java.util.HashTable | | |  | D. | |  | | --- | | Java.util.Collection | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Hash table based implementation of the Map interface.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-132) |

|  |  |
| --- | --- |
| 8. | What will be the output of the program?  package foo;  import java.util.Vector; /\* Line 2 \*/  private class MyVector extends Vector  {  int i = 1; /\* Line 5 \*/  public MyVector()  {  i = 2;  }  }  public class MyNewVector extends MyVector  {  public MyNewVector ()  {  i = 4; /\* Line 15 \*/  }  public static void main (String args [])  {  MyVector v = new MyNewVector(); /\* Line 19 \*/  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Compilation will succeed. | | |  | B. | |  | | --- | | Compilation will fail at line 3. | | |  | C. | |  | | --- | | Compilation will fail at line 5. | | |  | D. | |  | | --- | | Compilation will fail at line 15. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B is correct. The compiler complains with the error "modifier private not allowed here". The class is created private and is being used by another class on line 19.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-136) |

|  |  |
| --- | --- |
| 9. | What two statements are true about properly overridden hashCode() and equals() methods?   1. hashCode() doesn't have to be overridden if equals() is. 2. equals() doesn't have to be overridden if hashCode() is. 3. hashCode() can always return the same value, regardless of the object that invoked it. 4. equals() can be true even if it's comparing different objects. |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 2 | | |  | B. | |  | | --- | | 2 and 3 | | |  | C. | |  | | --- | | 3 and 4 | | |  | D. | |  | | --- | | 1 and 3 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  (3) and (4) are correct.  (1) and (2) are incorrect because by contract hashCode() and equals() can't be overridden unless both are overridden.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-143) |

|  |  |
| --- | --- |
| 10. | Which three are methods of the Object class?   1. notify(); 2. notifyAll(); 3. isInterrupted(); 4. synchronized(); 5. interrupt(); 6. wait(long msecs); 7. sleep(long msecs); 8. yield(); |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1, 2, 4 | | |  | B. | |  | | --- | | 2, 4, 5 | | |  | C. | |  | | --- | | 1, 2, 6 | | |  | D. | |  | | --- | | 2, 3, 4 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  (1), (2), and (6) are correct. They are all related to the list of threads waiting on the specified object.  (3), (5), (7), and (8) are incorrect answers. The methods isInterrupted() and interrupt() are instance methods of Thread.  The methods sleep() and yield() are static methods of Thread.  D is incorrect because synchronized is a keyword and the synchronized() construct is part of the Java language.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-173) |

|  |  |
| --- | --- |
| 11. | Which cannot directly cause a thread to stop executing? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Calling the SetPriority() method on a Thread object. | | |  | B. | |  | | --- | | Calling the wait() method on an object. | | |  | C. | |  | | --- | | Calling notify() method on an object. | | |  | D. | |  | | --- | | Calling read() method on an InputStream object. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Option C is correct. notify() - wakes up a single thread that is waiting on this object's monitor.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-168) |

|  |  |
| --- | --- |
| 12. | Which method registers a thread in a thread scheduler? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | run(); | | |  | B. | |  | | --- | | construct(); | | |  | C. | |  | | --- | | start(); | | |  | D. | |  | | --- | | register(); | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Option C is correct. The start() method causes this thread to begin execution; the Java Virtual Machine calls the run method of this thread.  Option A is wrong. The run() method of a thread is like the main() method to an application. Starting the thread causes the object's run method to be called in that separately executing thread.  Option B is wrong. There is no construct() method in the Thread class.  Option D is wrong. There is no register() method in the Thread class.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-166) |

|  |  |
| --- | --- |
| 13. | public class MyRunnable implements Runnable  {  public void run()  {  // some code here  }  }  which of these will create and start this thread? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | new Runnable(MyRunnable).start(); | | |  | B. | |  | | --- | | new Thread(MyRunnable).run(); | | |  | C. | |  | | --- | | new Thread(new MyRunnable()).start(); | | |  | D. | |  | | --- | | new MyRunnable().start(); | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Because the class implements Runnable, an instance of it has to be passed to the Thread constructor, and then the instance of the Thread has to be started.  A is incorrect. There is no constructor like this for Runnable because Runnable is an interface, and it is illegal to pass a class or interface name to any constructor.  B is incorrect for the same reason; you can't pass a class or interface name to any constructor.  D is incorrect because MyRunnable doesn't have a start() method, and the only start() method that can start a thread of execution is the start() in the Thread class.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-174) |

|  |  |
| --- | --- |
| 14. | void start() {  A a = new A();  B b = new B();  a.s(b);  b = null; /\* Line 5 \*/  a = null; /\* Line 6 \*/  System.out.println("start completed"); /\* Line 7 \*/  }  When is the B object, created in line 3, eligible for garbage collection? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | after line 5 | | |  | B. | |  | | --- | | after line 6 | | |  | C. | |  | | --- | | after line 7 | | |  | D. | |  | | --- | | There is no way to be absolutely certain. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Learn more problems on : [Garbage Collections](http://www.indiabix.com/java-programming/garbage-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/garbage-collections/discussion-206) |

|  |  |
| --- | --- |
| 15. | class X2  {  public X2 x;  public static void main(String [] args)  {  X2 x2 = new X2(); /\* Line 6 \*/  X2 x3 = new X2(); /\* Line 7 \*/  x2.x = x3;  x3.x = x2;  x2 = new X2();  x3 = x2; /\* Line 11 \*/  doComplexStuff();  }  }  after line 11 runs, how many objects are eligible for garbage collection? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 0 | | |  | B. | |  | | --- | | 1 | | |  | C. | |  | | --- | | 2 | | |  | D. | |  | | --- | | 3 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  This is an example of the islands of isolated objects. By the time line 11 has run, the objects instantiated in lines 6 and 7 are referring to each other, but no live thread can reach either of them.  Learn more problems on : [Garbage Collections](http://www.indiabix.com/java-programming/garbage-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/garbage-collections/discussion-203) |

|  |  |
| --- | --- |
| 16. | Which of the following would compile without error? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | int a = Math.abs(-5); | | |  | B. | |  | | --- | | int b = Math.abs(5.0); | | |  | C. | |  | | --- | | int c = Math.abs(5.5F); | | |  | D. | |  | | --- | | int d = Math.abs(5L); | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  The return value of the Math.abs() method is always the same as the type of the parameter passed into that method.  In the case of A, an integer is passed in and so the result is also an integer which is fine for assignment to "int a".  The values used in B, C & D respectively are a double, a float and a long. The compiler will complain about a possible loss of precision if we try to assign the results to an "int".  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-228) |

|  |  |
| --- | --- |
| 17. | What will be the output of the program?  String x = new String("xyz");  String y = "abc";  x = x + y;  How many String objects have been created? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 2 | | |  | B. | |  | | --- | | 3 | | |  | C. | |  | | --- | | 4 | | |  | D. | |  | | --- | | 5 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Line 1 creates two, one referred to by x and the lost String "xyz". Line 2 creates one (for a total of three). Line 3 creates one more (for a total of four), the concatenated String referred to by x with a value of "xyzabc".  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-233) |

|  |  |
| --- | --- |
| 18. | What will be the output of the program?  System.out.println(Math.sqrt(-4D)); |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | -2 | | |  | B. | |  | | --- | | NaN | | |  | C. | |  | | --- | | Compile Error | | |  | D. | |  | | --- | | Runtime Exception | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  It is not possible in regular mathematics to get a value for the square-root of a negative number therefore a NaN will be returned because the code is valid.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-244) |

|  |  |
| --- | --- |
| 19. | What will be the output of the program?  public class Test  {  public static void main(String[] args)  {  final StringBuffer a = new StringBuffer();  final StringBuffer b = new StringBuffer();  new Thread()  {  public void run()  {  System.out.print(a.append("A"));  synchronized(b)  {  System.out.print(b.append("B"));  }  }  }.start();    new Thread()  {  public void run()  {  System.out.print(b.append("C"));  synchronized(a)  {  System.out.print(a.append("D"));  }  }  }.start();  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | ACCBAD | | |  | B. | |  | | --- | | ABBCAD | | |  | C. | |  | | --- | | CDDACB | | |  | D. | |  | | --- | | Indeterminate output | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  It gives different output while executing the same compiled code at different times.  C:\>javac Test.java  C:\>java Test  ABBCAD  C:\>java Test  ACADCB  C:\>java Test  ACBCBAD  C:\>java Test  ABBCAD  C:\>java Test  ACBCBAD  C:\>java Test  ACBCBAD  C:\>java Test  ABBCAD  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-251) |

|  |  |
| --- | --- |
| 20. | Which statement is true given the following?  Double d = Math.random(); |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 0.0 < d <= 1.0 | | |  | B. | |  | | --- | | 0.0 <= d < 1.0 | | |  | C. | |  | | --- | | Compilation fail | | |  | D. | |  | | --- | | Cannot say. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  The Math.random() method returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-262) |

# Java Programming Test 8

[Home](http://www.indiabix.com/) » [Online Test](http://www.indiabix.com/online-test/categories/) » [Online Java Programming Test](http://www.indiabix.com/online-test/java-programming-test/) » Java Programming Test 8

|  |  |  |
| --- | --- | --- |
| **Marks : 0/20** | | |
| Total number of questions | : | **20** |
| Number of answered questions | : | **0** |
| Number of unanswered questions | : | **20** |

### Test Review : View answers and explanation for this test.

|  |  |
| --- | --- |
| 1. | Which two of the following are legal declarations for nonnested classes and interfaces?   1. final abstract class Test {} 2. public static interface Test {} 3. final public class Test {} 4. protected abstract class Test {} 5. protected interface Test {} 6. abstract public class Test {} |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 4 | | |  | B. | |  | | --- | | 2 and 5 | | |  | C. | |  | | --- | | 3 and 6 | | |  | D. | |  | | --- | | 4 and 6 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  (3), (6). Both are legal class declarations.  (1) is wrong because a class cannot be abstract and finalâ€”there would be no way to use such a class. (2) is wrong because interfaces and classes cannot be marked as static. (4) and (5) are wrong because classes and interfaces cannot be marked as protected.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-26) |

|  |  |
| --- | --- |
| 2. | public class While  {  public void loop()  {  int x= 0;  while ( 1 ) /\* Line 6 \*/  {  System.out.print("x plus one is " + (x + 1)); /\* Line 8 \*/  }  }  }  Which statement is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | There is a syntax error on line 1. | | |  | B. | |  | | --- | | There are syntax errors on lines 1 and 6. | | |  | C. | |  | | --- | | There are syntax errors on lines 1, 6, and 8. | | |  | D. | |  | | --- | | There is a syntax error on line 6. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Using the integer 1 in the while statement, or any other looping or conditional construct for that matter, will result in a compiler error. This is old C Program syntax, not valid Java.  A, B and C are incorrect because line 1 is valid (Java is case sensitive so While is a valid class name). Line 8 is also valid because an equation may be placed in a String operation as shown.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-81) |

|  |  |
| --- | --- |
| 3. | What will be the output of the program?  int I = 0;  outer:  while (true)  {  I++;  inner:  for (int j = 0; j < 10; j++)  {  I += j;  if (j == 3)  continue inner;  break outer;  }  continue outer;  }  System.out.println(I); |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 | | |  | B. | |  | | --- | | 2 | | |  | C. | |  | | --- | | 3 | | |  | D. | |  | | --- | | 4 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  The program flows as follows: I will be incremented after the while loop is entered, then I will be incremented (by zero) when the for loop is entered. The if statement evaluates to false, and the continuestatement is never reached. The break statement tells the JVM to break out of the outer loop, at which point I is printed and the fragment is done.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-89) |

|  |  |
| --- | --- |
| 4. | What will be the output of the program?  public class Test  {  private static float[] f = new float[2];  public static void main (String[] args)  {  System.out.println("f[0] = " + f[0]);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | f[0] = 0 | | |  | B. | |  | | --- | | f[0] = 0.0 | | |  | C. | |  | | --- | | Compile Error | | |  | D. | |  | | --- | | Runtime Exception | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  The choices are between Option A and B, what this question is really testing is your knowledge of default values of an initialized array. This is an array type float i.e. it is a type that uses decimal point numbers therefore its initial value will be 0.0 and not 0  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-140) |

|  |  |
| --- | --- |
| 5. | Which two statements are true about comparing two instances of the same class, given that the equals() and hashCode() methods have been properly overridden?   1. If the equals() method returns true, the hashCode() comparison == must return true. 2. If the equals() method returns false, the hashCode() comparison != must return true. 3. If the hashCode() comparison == returns true, the equals() method must return true. 4. If the hashCode() comparison == returns true, the equals() method might return true. |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 4 | | |  | B. | |  | | --- | | 2 and 3 | | |  | C. | |  | | --- | | 3 and 4 | | |  | D. | |  | | --- | | 1 and 3 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  (1) is a restatement of the equals() and hashCode() contract. (4) is true because if the hashCode()comparison returns ==, the two objects might or might not be equal.  (2) and (3) are incorrect because the hashCode() method is very flexible in its return values, and often two dissimilar objects can return the same hash code value.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-147) |

|  |  |
| --- | --- |
| 6. | x = 0;  if (x1.hashCode() != x2.hashCode() ) x = x + 1;  if (x3.equals(x4) ) x = x + 10;  if (!x5.equals(x6) ) x = x + 100;  if (x7.hashCode() == x8.hashCode() ) x = x + 1000;  System.out.println("x = " + x);  and assuming that the equals() and hashCode() methods are properly implemented, if the output is "x = 1111", which of the following statements will always be true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | x2.equals(x1) | | |  | B. | |  | | --- | | x3.hashCode() == x4.hashCode() | | |  | C. | |  | | --- | | x5.hashCode() != x6.hashCode() | | |  | D. | |  | | --- | | x8.equals(x7) | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  By contract, if two objects are equivalent according to the equals() method, then the hashCode() method must evaluate them to be ==.  Option A is incorrect because if the hashCode() values are not equal, the two objects must not be equal.  Option C is incorrect because if equals() is not true there is no guarantee of any result from hashCode().  Option D is incorrect because hashCode() will often return == even if the two objects do not evaluate to equals() being true.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-150) |

|  |  |
| --- | --- |
| 7. | class Boo  {  Boo(String s) { }  Boo() { }  }  class Bar extends Boo  {  Bar() { }  Bar(String s) {super(s);}  void zoo()  {  // insert code here  }  }  which one create an anonymous inner class from within class Bar? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Boo f = new Boo(24) { }; | | |  | B. | |  | | --- | | Boo f = new Bar() { }; | | |  | C. | |  | | --- | | Bar f = new Boo(String s) { }; | | |  | D. | |  | | --- | | Boo f = new Boo.Bar(String s) { }; | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B is correct because anonymous inner classes are no different from any other class when it comes to polymorphism. That means you are always allowed to declare a reference variable of the superclass type and have that reference variable refer to an instance of a subclass type, which in this case is an anonymous subclass of Bar. Since Bar is a subclass of Boo, it all works.  Option A is incorrect because it passes an int to the Boo constructor, and there is no matching constructor in the Boo class.  Option C is incorrect because it violates the rules of polymorphismâ€”you cannot refer to a superclass type using a reference variable declared as the subclass type. The superclass is not guaranteed to have everything the subclass has.  Option D uses incorrect syntax.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-154) |

|  |  |
| --- | --- |
| 8. | Which statement is true about a static nested class? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | You must have a reference to an instance of the enclosing class in order to instantiate it. | | |  | B. | |  | | --- | | It does not have access to nonstatic members of the enclosing class. | | |  | C. | |  | | --- | | It's variables and methods must be static. | | |  | D. | |  | | --- | | It must extend the enclosing class. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B is correct because a static nested class is not tied to an instance of the enclosing class, and thus can't access the nonstatic members of the class (just as a static method can't access nonstatic members of a class).  Option A is incorrect because static nested classes do not need (and can't use) a reference to an instance of the enclosing class.  Option C is incorrect because static nested classes can declare and define nonstatic members.  Option D is wrong because it just is. There's no rule that says an inner or nested class has to extend anything.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-152) |

|  |  |
| --- | --- |
| 9. | Which constructs an anonymous inner class instance? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Runnable r = new Runnable() { }; | | |  | B. | |  | | --- | | Runnable r = new Runnable(public void run() { }); | | |  | C. | |  | | --- | | Runnable r = new Runnable { public void run(){}}; | | |  | D. | |  | | --- | | System.out.println(new Runnable() {public void run() { }}); | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  D is correct. It defines an anonymous inner class instance, which also means it creates an instance of that new anonymous class at the same time. The anonymous class is an implementer of the Runnable interface, so it must override the run() method of Runnable.  A is incorrect because it doesn't override the run() method, so it violates the rules of interface implementation.  B and C use incorrect syntax.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-153) |

|  |  |
| --- | --- |
| 10. | What will be the output of the program?  public class HorseTest  {  public static void main (String [] args)  {  class Horse  {  public String name; /\* Line 7 \*/  public Horse(String s)  {  name = s;  }  } /\* class Horse ends \*/    Object obj = new Horse("Zippo"); /\* Line 13 \*/  Horse h = (Horse) obj; /\* Line 14 \*/  System.out.println(h.name);  }  } /\* class HorseTest ends \*/ |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | An exception occurs at runtime at line 10. | | |  | B. | |  | | --- | | It prints "Zippo". | | |  | C. | |  | | --- | | Compilation fails because of an error on line 7. | | |  | D. | |  | | --- | | Compilation fails because of an error on line 13. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  The code in the HorseTest class is perfectly legal. Line 13 creates an instance of the method-local inner class Horse, using a reference variable declared as type Object. Line 14 casts the Horse object to a Horsereference variable, which allows line 15 to compile. If line 14 were removed, the HorseTest code would not compile, because class Object does not have a name variable.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-159) |

|  |  |
| --- | --- |
| 11. | What will be the output of the program?  public abstract class AbstractTest  {  public int getNum()  {  return 45;  }  public abstract class Bar  {  public int getNum()  {  return 38;  }  }  public static void main (String [] args)  {  AbstractTest t = new AbstractTest()  {  public int getNum()  {  return 22;  }  };  AbstractTest.Bar f = t.new Bar()  {  public int getNum()  {  return 57;  }  };    System.out.println(f.getNum() + " " + t.getNum());  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 57 22 | | |  | B. | |  | | --- | | 45 38 | | |  | C. | |  | | --- | | 45 57 | | |  | D. | |  | | --- | | An exception occurs at runtime. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  You can define an inner class as abstract, which means you can instantiate only concrete subclasses of the abstract inner class. The object referenced by the variable t is an instance of an anonymous subclass of AbstractTest, and the anonymous class overrides the getNum() method to return 22. The variable referenced by f is an instance of an anonymous subclass of Bar, and the anonymous Bar subclass also overrides the getNum() method (to return 57). Remember that to instantiate a Bar instance, we need an instance of the enclosing AbstractTest class to tie to the new Bar inner class instance. AbstractTest can't be instantiated because it's abstract, so we created an anonymous subclass (non-abstract) and then used the instance of that anonymous subclass to tie to the new Bar subclass instance.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-160) |

|  |  |
| --- | --- |
| 12. | What will be the output of the program?  public class Q126 implements Runnable  {  private int x;  private int y;  public static void main(String [] args)  {  Q126 that = new Q126();  (new Thread(that)).start( ); /\* Line 8 \*/  (new Thread(that)).start( ); /\* Line 9 \*/  }  public synchronized void run( ) /\* Line 11 \*/  {  for (;;) /\* Line 13 \*/  {  x++;  y++;  System.out.println("x = " + x + "y = " + y);  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | An error at line 11 causes compilation to fail | | |  | B. | |  | | --- | | Errors at lines 8 and 9 cause compilation to fail. | | |  | C. | |  | | --- | | The program prints pairs of values for x and y that might not always be the same on the same line (for example, "x=2, y=1") | | |  | D. | |  | | --- | | The program prints pairs of values for x and y that are always the same on the same line (for example, "x=1, y=1". In addition, each value appears once (for example, "x=1, y=1" followed by "x=2, y=2") | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  The synchronized code is the key to answering this question. Because x and y are both incremented inside the synchronized method they are always incremented together. Also keep in mind that the two threads share the same reference to the Q126 object.  Also note that because of the infinite loop at line 13, only one thread ever gets to execute.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-191) |

|  |  |
| --- | --- |
| 13. | What will be the output of the program?  class MyThread extends Thread  {  public static void main(String [] args)  {  MyThread t = new MyThread(); /\* Line 5 \*/  t.run(); /\* Line 6 \*/  }  public void run()  {  for(int i=1; i < 3; ++i)  {  System.out.print(i + "..");  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | This code will not compile due to line 5. | | |  | B. | |  | | --- | | This code will not compile due to line 6. | | |  | C. | |  | | --- | | 1..2.. | | |  | D. | |  | | --- | | 1..2..3.. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Line 6 calls the run() method, so the run() method executes as a normal method should and it prints "1..2.."  A is incorrect because line 5 is the proper way to create an object.  B is incorrect because it is legal to call the run() method, even though this will not start a true thread of execution. The code after line 6 will not execute until the run() method is complete.  D is incorrect because the for loop only does two iterations.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-178) |

|  |  |
| --- | --- |
| 14. | What will be the output of the program?  public class ThreadTest extends Thread  {  public void run()  {  System.out.println("In run");  yield();  System.out.println("Leaving run");  }  public static void main(String []argv)  {  (new ThreadTest()).start();  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | The code fails to compile in the main() method | | |  | B. | |  | | --- | | The code fails to compile in the run() method | | |  | C. | |  | | --- | | Only the text "In run" will be displayed | | |  | D. | |  | | --- | | The text "In run" followed by "Leaving run" will be displayed | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-182) |

|  |  |
| --- | --- |
| 15. | What will be the output of the program?  public class Test107 implements Runnable  {  private int x;  private int y;  public static void main(String args[])  {  Test107 that = new Test107();  (new Thread(that)).start();  (new Thread(that)).start();  }  public synchronized void run()  {  for(int i = 0; i < 10; i++)  {  x++;  y++;  System.out.println("x = " + x + ", y = " + y); /\* Line 17 \*/  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Compilation error. | | |  | B. | |  | | --- | | Will print in this order: x = 1 y = 1 x = 2 y = 2 x = 3 y = 3 x = 4 y = 4 x = 5 y = 5...but the output will be produced by both threads running simultaneously. | | |  | C. | |  | | --- | | Will print in this order: x = 1 y = 1 x = 2 y = 2 x = 3 y = 3 x = 4 y = 4 x = 5 y = 5...but the output will be produced by first one thread then the other. This is guaranteed by the synchronised code. | | |  | D. | |  | | --- | | Will print in this order x = 1 y = 2 x = 3 y = 4 x = 5 y = 6 x = 7 y = 8... | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Both threads are operating on the same instance variables. Because the code is synchronized the first thread will complete before the second thread begins. Modify line 17 to print the thread names:  System.out.println(Thread.currentThread().getName() + " x = " + x + ", y = " + y);  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-185) |

|  |  |
| --- | --- |
| 16. | What allows the programmer to destroy an object x? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | x.delete() | | |  | B. | |  | | --- | | x.finalize() | | |  | C. | |  | | --- | | Runtime.getRuntime().gc() | | |  | D. | |  | | --- | | Only the garbage collection system can destroy an object. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Option D is correct. When an object is no longer referenced, it may be reclaimed by the garbage collector. If an object declares a finalizer, the finalizer is executed before the object is reclaimed to give the object a last chance to clean up resources that would not otherwise be released. When a class is no longer needed, it may be unloaded.  Option A is wrong. I found 4 delete() methods in all of the Java class structure. They are:   1. delete() - Method in class java.io.File : Deletes the file or directory denoted by this abstract pathname. 2. delete(int, int) - Method in class java.lang.StringBuffer : Removes the characters in a substring of this StringBuffer. 3. delete(int, int) - Method in interface javax.accessibility.AccessibleEditableText : Deletes the text between two indices 4. delete(int, int) - Method in class : javax.swing.text.JTextComponent.AccessibleJTextComponent; Deletes the text between two indices   None of these destroy the object to which they belong.  Option B is wrong. I found 19 finalize() methods. The most interesting, from this questions point of view, was the finalize() method in class java.lang.Object which is called by the garbage collector on an object when garbage collection determines that there are no more references to the object. This method does not destroy the object to which it belongs.  Option C is wrong. But it is interesting. The Runtime class has many methods, two of which are:   1. getRuntime() - Returns the runtime object associated with the current Java application. 2. gc() - Runs the garbage collector. Calling this method suggests that the Java virtual machine expend effort toward recycling unused objects in order to make the memory they currently occupy available for quick reuse. When control returns from the method call, the virtual machine has made its best effort to recycle all discarded objects. Interesting as this is, it doesn't destroy the object.   Learn more problems on : [Garbage Collections](http://www.indiabix.com/java-programming/garbage-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/garbage-collections/discussion-204) |

|  |  |
| --- | --- |
| 17. | Which of the following statements is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | It is sometimes good practice to throw an AssertionError explicitly. | | |  | B. | |  | | --- | | Private getter() and setter() methods should not use assertions to verify arguments. | | |  | C. | |  | | --- | | If an AssertionError is thrown in a try-catch block, the finally block will be bypassed. | | |  | D. | |  | | --- | | It is proper to handle assertion statement failures using a catch (AssertionException ae) block. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Option A is correct because it is sometimes advisable to thrown an assertion error even if assertions have been disabled.  Option B is incorrect because it is considered appropriate to check argument values in private methods using assertions.  Option C is incorrect; finally is never bypassed.  Option D is incorrect because AssertionErrors should never be handled.  Learn more problems on : [Assertions](http://www.indiabix.com/java-programming/assertions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/assertions/discussion-220) |

|  |  |
| --- | --- |
| 18. | What will be the output of the program?  public class WrapTest  {  public static void main(String [] args)  {  int result = 0;  short s = 42;  Long x = new Long("42");  Long y = new Long(42);  Short z = new Short("42");  Short x2 = new Short(s);  Integer y2 = new Integer("42");  Integer z2 = new Integer(42);  if (x == y) /\* Line 13 \*/  result = 1;  if (x.equals(y) ) /\* Line 15 \*/  result = result + 10;  if (x.equals(z) ) /\* Line 17 \*/  result = result + 100;  if (x.equals(x2) ) /\* Line 19 \*/  result = result + 1000;  if (x.equals(z2) ) /\* Line 21 \*/  result = result + 10000;  System.out.println("result = " + result);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | result = 1 | | |  | B. | |  | | --- | | result = 10 | | |  | C. | |  | | --- | | result = 11 | | |  | D. | |  | | --- | | result = 11010 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Line 13 fails because == compares reference values, not object values. Line 15 succeeds because both String and primitive wrapper constructors resolve to the same value (except for the Character wrapper). Lines 17, 19, and 21 fail because the equals() method fails if the object classes being compared are different and not in the same tree hierarchy.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-241) |

|  |  |
| --- | --- |
| 19. | What will be the output of the program?  class Q207  {  public static void main(String[] args)  {  int i1 = 5;  int i2 = 6;  String s1 = "7";  System.out.println(i1 + i2 + s1); /\* Line 8 \*/  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 18 | | |  | B. | |  | | --- | | 117 | | |  | C. | |  | | --- | | 567 | | |  | D. | |  | | --- | | Compiler error | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  This question is about the + (plus) operator and the overriden + (string cocatanation) operator. The rules that apply when you have a mixed expression of numbers and strings are:  If either operand is a String, the + operator concatenates the operands.  If both operands are numeric, the + operator adds the operands.  The expression on line 6 above can be read as "Add the values i1 and i2 together, then take the sum and convert it to a string and concatenate it with the String from the variable s1". In code, the compiler probably interprets the expression on line 8 above as:  System.out.println( new StringBuffer()  .append(new Integer(i1 + i2).toString())  .append(s1)  .toString() );  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-257) |

|  |  |
| --- | --- |
| 20. | What will be the output of the program?  public class StringRef  {  public static void main(String [] args)  {  String s1 = "abc";  String s2 = "def";  String s3 = s2; /\* Line 7 \*/  s2 = "ghi";  System.out.println(s1 + s2 + s3);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | abcdefghi | | |  | B. | |  | | --- | | abcdefdef | | |  | C. | |  | | --- | | abcghidef | | |  | D. | |  | | --- | | abcghighi | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  After line 7 executes, both s2 and s3 refer to a String object that contains the value "def". When line 8 executes, a new String object is created with the value "ghi", to which s2 refers. The reference variable s3still refers to the (immutable) String object with the value "def".  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-231) |

# Java Programming Test 8

[Home](http://www.indiabix.com/) » [Online Test](http://www.indiabix.com/online-test/categories/) » [Online Java Programming Test](http://www.indiabix.com/online-test/java-programming-test/) » Java Programming Test 8

|  |  |  |
| --- | --- | --- |
| **Marks : 0/20** | | |
| Total number of questions | : | **20** |
| Number of answered questions | : | **0** |
| Number of unanswered questions | : | **20** |

### Test Review : View answers and explanation for this test.

|  |  |
| --- | --- |
| 1. | Which two of the following are legal declarations for nonnested classes and interfaces?   1. final abstract class Test {} 2. public static interface Test {} 3. final public class Test {} 4. protected abstract class Test {} 5. protected interface Test {} 6. abstract public class Test {} |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 4 | | |  | B. | |  | | --- | | 2 and 5 | | |  | C. | |  | | --- | | 3 and 6 | | |  | D. | |  | | --- | | 4 and 6 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  (3), (6). Both are legal class declarations.  (1) is wrong because a class cannot be abstract and finalâ€”there would be no way to use such a class. (2) is wrong because interfaces and classes cannot be marked as static. (4) and (5) are wrong because classes and interfaces cannot be marked as protected.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-26) |

|  |  |
| --- | --- |
| 2. | public class While  {  public void loop()  {  int x= 0;  while ( 1 ) /\* Line 6 \*/  {  System.out.print("x plus one is " + (x + 1)); /\* Line 8 \*/  }  }  }  Which statement is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | There is a syntax error on line 1. | | |  | B. | |  | | --- | | There are syntax errors on lines 1 and 6. | | |  | C. | |  | | --- | | There are syntax errors on lines 1, 6, and 8. | | |  | D. | |  | | --- | | There is a syntax error on line 6. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Using the integer 1 in the while statement, or any other looping or conditional construct for that matter, will result in a compiler error. This is old C Program syntax, not valid Java.  A, B and C are incorrect because line 1 is valid (Java is case sensitive so While is a valid class name). Line 8 is also valid because an equation may be placed in a String operation as shown.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-81) |

|  |  |
| --- | --- |
| 3. | What will be the output of the program?  int I = 0;  outer:  while (true)  {  I++;  inner:  for (int j = 0; j < 10; j++)  {  I += j;  if (j == 3)  continue inner;  break outer;  }  continue outer;  }  System.out.println(I); |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 | | |  | B. | |  | | --- | | 2 | | |  | C. | |  | | --- | | 3 | | |  | D. | |  | | --- | | 4 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  The program flows as follows: I will be incremented after the while loop is entered, then I will be incremented (by zero) when the for loop is entered. The if statement evaluates to false, and the continuestatement is never reached. The break statement tells the JVM to break out of the outer loop, at which point I is printed and the fragment is done.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-89) |

|  |  |
| --- | --- |
| 4. | What will be the output of the program?  public class Test  {  private static float[] f = new float[2];  public static void main (String[] args)  {  System.out.println("f[0] = " + f[0]);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | f[0] = 0 | | |  | B. | |  | | --- | | f[0] = 0.0 | | |  | C. | |  | | --- | | Compile Error | | |  | D. | |  | | --- | | Runtime Exception | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  The choices are between Option A and B, what this question is really testing is your knowledge of default values of an initialized array. This is an array type float i.e. it is a type that uses decimal point numbers therefore its initial value will be 0.0 and not 0  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-140) |

|  |  |
| --- | --- |
| 5. | Which two statements are true about comparing two instances of the same class, given that the equals() and hashCode() methods have been properly overridden?   1. If the equals() method returns true, the hashCode() comparison == must return true. 2. If the equals() method returns false, the hashCode() comparison != must return true. 3. If the hashCode() comparison == returns true, the equals() method must return true. 4. If the hashCode() comparison == returns true, the equals() method might return true. |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 4 | | |  | B. | |  | | --- | | 2 and 3 | | |  | C. | |  | | --- | | 3 and 4 | | |  | D. | |  | | --- | | 1 and 3 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  (1) is a restatement of the equals() and hashCode() contract. (4) is true because if the hashCode()comparison returns ==, the two objects might or might not be equal.  (2) and (3) are incorrect because the hashCode() method is very flexible in its return values, and often two dissimilar objects can return the same hash code value.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-147) |

|  |  |
| --- | --- |
| 6. | x = 0;  if (x1.hashCode() != x2.hashCode() ) x = x + 1;  if (x3.equals(x4) ) x = x + 10;  if (!x5.equals(x6) ) x = x + 100;  if (x7.hashCode() == x8.hashCode() ) x = x + 1000;  System.out.println("x = " + x);  and assuming that the equals() and hashCode() methods are properly implemented, if the output is "x = 1111", which of the following statements will always be true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | x2.equals(x1) | | |  | B. | |  | | --- | | x3.hashCode() == x4.hashCode() | | |  | C. | |  | | --- | | x5.hashCode() != x6.hashCode() | | |  | D. | |  | | --- | | x8.equals(x7) | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  By contract, if two objects are equivalent according to the equals() method, then the hashCode() method must evaluate them to be ==.  Option A is incorrect because if the hashCode() values are not equal, the two objects must not be equal.  Option C is incorrect because if equals() is not true there is no guarantee of any result from hashCode().  Option D is incorrect because hashCode() will often return == even if the two objects do not evaluate to equals() being true.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-150) |

|  |  |
| --- | --- |
| 7. | class Boo  {  Boo(String s) { }  Boo() { }  }  class Bar extends Boo  {  Bar() { }  Bar(String s) {super(s);}  void zoo()  {  // insert code here  }  }  which one create an anonymous inner class from within class Bar? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Boo f = new Boo(24) { }; | | |  | B. | |  | | --- | | Boo f = new Bar() { }; | | |  | C. | |  | | --- | | Bar f = new Boo(String s) { }; | | |  | D. | |  | | --- | | Boo f = new Boo.Bar(String s) { }; | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B is correct because anonymous inner classes are no different from any other class when it comes to polymorphism. That means you are always allowed to declare a reference variable of the superclass type and have that reference variable refer to an instance of a subclass type, which in this case is an anonymous subclass of Bar. Since Bar is a subclass of Boo, it all works.  Option A is incorrect because it passes an int to the Boo constructor, and there is no matching constructor in the Boo class.  Option C is incorrect because it violates the rules of polymorphismâ€”you cannot refer to a superclass type using a reference variable declared as the subclass type. The superclass is not guaranteed to have everything the subclass has.  Option D uses incorrect syntax.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-154) |

|  |  |
| --- | --- |
| 8. | Which statement is true about a static nested class? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | You must have a reference to an instance of the enclosing class in order to instantiate it. | | |  | B. | |  | | --- | | It does not have access to nonstatic members of the enclosing class. | | |  | C. | |  | | --- | | It's variables and methods must be static. | | |  | D. | |  | | --- | | It must extend the enclosing class. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B is correct because a static nested class is not tied to an instance of the enclosing class, and thus can't access the nonstatic members of the class (just as a static method can't access nonstatic members of a class).  Option A is incorrect because static nested classes do not need (and can't use) a reference to an instance of the enclosing class.  Option C is incorrect because static nested classes can declare and define nonstatic members.  Option D is wrong because it just is. There's no rule that says an inner or nested class has to extend anything.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-152) |

|  |  |
| --- | --- |
| 9. | Which constructs an anonymous inner class instance? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Runnable r = new Runnable() { }; | | |  | B. | |  | | --- | | Runnable r = new Runnable(public void run() { }); | | |  | C. | |  | | --- | | Runnable r = new Runnable { public void run(){}}; | | |  | D. | |  | | --- | | System.out.println(new Runnable() {public void run() { }}); | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  D is correct. It defines an anonymous inner class instance, which also means it creates an instance of that new anonymous class at the same time. The anonymous class is an implementer of the Runnable interface, so it must override the run() method of Runnable.  A is incorrect because it doesn't override the run() method, so it violates the rules of interface implementation.  B and C use incorrect syntax.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-153) |

|  |  |
| --- | --- |
| 10. | What will be the output of the program?  public class HorseTest  {  public static void main (String [] args)  {  class Horse  {  public String name; /\* Line 7 \*/  public Horse(String s)  {  name = s;  }  } /\* class Horse ends \*/    Object obj = new Horse("Zippo"); /\* Line 13 \*/  Horse h = (Horse) obj; /\* Line 14 \*/  System.out.println(h.name);  }  } /\* class HorseTest ends \*/ |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | An exception occurs at runtime at line 10. | | |  | B. | |  | | --- | | It prints "Zippo". | | |  | C. | |  | | --- | | Compilation fails because of an error on line 7. | | |  | D. | |  | | --- | | Compilation fails because of an error on line 13. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  The code in the HorseTest class is perfectly legal. Line 13 creates an instance of the method-local inner class Horse, using a reference variable declared as type Object. Line 14 casts the Horse object to a Horsereference variable, which allows line 15 to compile. If line 14 were removed, the HorseTest code would not compile, because class Object does not have a name variable.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-159) |

|  |  |
| --- | --- |
| 11. | What will be the output of the program?  public abstract class AbstractTest  {  public int getNum()  {  return 45;  }  public abstract class Bar  {  public int getNum()  {  return 38;  }  }  public static void main (String [] args)  {  AbstractTest t = new AbstractTest()  {  public int getNum()  {  return 22;  }  };  AbstractTest.Bar f = t.new Bar()  {  public int getNum()  {  return 57;  }  };    System.out.println(f.getNum() + " " + t.getNum());  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 57 22 | | |  | B. | |  | | --- | | 45 38 | | |  | C. | |  | | --- | | 45 57 | | |  | D. | |  | | --- | | An exception occurs at runtime. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  You can define an inner class as abstract, which means you can instantiate only concrete subclasses of the abstract inner class. The object referenced by the variable t is an instance of an anonymous subclass of AbstractTest, and the anonymous class overrides the getNum() method to return 22. The variable referenced by f is an instance of an anonymous subclass of Bar, and the anonymous Bar subclass also overrides the getNum() method (to return 57). Remember that to instantiate a Bar instance, we need an instance of the enclosing AbstractTest class to tie to the new Bar inner class instance. AbstractTest can't be instantiated because it's abstract, so we created an anonymous subclass (non-abstract) and then used the instance of that anonymous subclass to tie to the new Bar subclass instance.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-160) |

|  |  |
| --- | --- |
| 12. | What will be the output of the program?  public class Q126 implements Runnable  {  private int x;  private int y;  public static void main(String [] args)  {  Q126 that = new Q126();  (new Thread(that)).start( ); /\* Line 8 \*/  (new Thread(that)).start( ); /\* Line 9 \*/  }  public synchronized void run( ) /\* Line 11 \*/  {  for (;;) /\* Line 13 \*/  {  x++;  y++;  System.out.println("x = " + x + "y = " + y);  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | An error at line 11 causes compilation to fail | | |  | B. | |  | | --- | | Errors at lines 8 and 9 cause compilation to fail. | | |  | C. | |  | | --- | | The program prints pairs of values for x and y that might not always be the same on the same line (for example, "x=2, y=1") | | |  | D. | |  | | --- | | The program prints pairs of values for x and y that are always the same on the same line (for example, "x=1, y=1". In addition, each value appears once (for example, "x=1, y=1" followed by "x=2, y=2") | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  The synchronized code is the key to answering this question. Because x and y are both incremented inside the synchronized method they are always incremented together. Also keep in mind that the two threads share the same reference to the Q126 object.  Also note that because of the infinite loop at line 13, only one thread ever gets to execute.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-191) |

|  |  |
| --- | --- |
| 13. | What will be the output of the program?  class MyThread extends Thread  {  public static void main(String [] args)  {  MyThread t = new MyThread(); /\* Line 5 \*/  t.run(); /\* Line 6 \*/  }  public void run()  {  for(int i=1; i < 3; ++i)  {  System.out.print(i + "..");  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | This code will not compile due to line 5. | | |  | B. | |  | | --- | | This code will not compile due to line 6. | | |  | C. | |  | | --- | | 1..2.. | | |  | D. | |  | | --- | | 1..2..3.. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Line 6 calls the run() method, so the run() method executes as a normal method should and it prints "1..2.."  A is incorrect because line 5 is the proper way to create an object.  B is incorrect because it is legal to call the run() method, even though this will not start a true thread of execution. The code after line 6 will not execute until the run() method is complete.  D is incorrect because the for loop only does two iterations.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-178) |

|  |  |
| --- | --- |
| 14. | What will be the output of the program?  public class ThreadTest extends Thread  {  public void run()  {  System.out.println("In run");  yield();  System.out.println("Leaving run");  }  public static void main(String []argv)  {  (new ThreadTest()).start();  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | The code fails to compile in the main() method | | |  | B. | |  | | --- | | The code fails to compile in the run() method | | |  | C. | |  | | --- | | Only the text "In run" will be displayed | | |  | D. | |  | | --- | | The text "In run" followed by "Leaving run" will be displayed | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-182) |

|  |  |
| --- | --- |
| 15. | What will be the output of the program?  public class Test107 implements Runnable  {  private int x;  private int y;  public static void main(String args[])  {  Test107 that = new Test107();  (new Thread(that)).start();  (new Thread(that)).start();  }  public synchronized void run()  {  for(int i = 0; i < 10; i++)  {  x++;  y++;  System.out.println("x = " + x + ", y = " + y); /\* Line 17 \*/  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Compilation error. | | |  | B. | |  | | --- | | Will print in this order: x = 1 y = 1 x = 2 y = 2 x = 3 y = 3 x = 4 y = 4 x = 5 y = 5...but the output will be produced by both threads running simultaneously. | | |  | C. | |  | | --- | | Will print in this order: x = 1 y = 1 x = 2 y = 2 x = 3 y = 3 x = 4 y = 4 x = 5 y = 5...but the output will be produced by first one thread then the other. This is guaranteed by the synchronised code. | | |  | D. | |  | | --- | | Will print in this order x = 1 y = 2 x = 3 y = 4 x = 5 y = 6 x = 7 y = 8... | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Both threads are operating on the same instance variables. Because the code is synchronized the first thread will complete before the second thread begins. Modify line 17 to print the thread names:  System.out.println(Thread.currentThread().getName() + " x = " + x + ", y = " + y);  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-185) |

|  |  |
| --- | --- |
| 16. | What allows the programmer to destroy an object x? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | x.delete() | | |  | B. | |  | | --- | | x.finalize() | | |  | C. | |  | | --- | | Runtime.getRuntime().gc() | | |  | D. | |  | | --- | | Only the garbage collection system can destroy an object. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Option D is correct. When an object is no longer referenced, it may be reclaimed by the garbage collector. If an object declares a finalizer, the finalizer is executed before the object is reclaimed to give the object a last chance to clean up resources that would not otherwise be released. When a class is no longer needed, it may be unloaded.  Option A is wrong. I found 4 delete() methods in all of the Java class structure. They are:   1. delete() - Method in class java.io.File : Deletes the file or directory denoted by this abstract pathname. 2. delete(int, int) - Method in class java.lang.StringBuffer : Removes the characters in a substring of this StringBuffer. 3. delete(int, int) - Method in interface javax.accessibility.AccessibleEditableText : Deletes the text between two indices 4. delete(int, int) - Method in class : javax.swing.text.JTextComponent.AccessibleJTextComponent; Deletes the text between two indices   None of these destroy the object to which they belong.  Option B is wrong. I found 19 finalize() methods. The most interesting, from this questions point of view, was the finalize() method in class java.lang.Object which is called by the garbage collector on an object when garbage collection determines that there are no more references to the object. This method does not destroy the object to which it belongs.  Option C is wrong. But it is interesting. The Runtime class has many methods, two of which are:   1. getRuntime() - Returns the runtime object associated with the current Java application. 2. gc() - Runs the garbage collector. Calling this method suggests that the Java virtual machine expend effort toward recycling unused objects in order to make the memory they currently occupy available for quick reuse. When control returns from the method call, the virtual machine has made its best effort to recycle all discarded objects. Interesting as this is, it doesn't destroy the object.   Learn more problems on : [Garbage Collections](http://www.indiabix.com/java-programming/garbage-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/garbage-collections/discussion-204) |

|  |  |
| --- | --- |
| 17. | Which of the following statements is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | It is sometimes good practice to throw an AssertionError explicitly. | | |  | B. | |  | | --- | | Private getter() and setter() methods should not use assertions to verify arguments. | | |  | C. | |  | | --- | | If an AssertionError is thrown in a try-catch block, the finally block will be bypassed. | | |  | D. | |  | | --- | | It is proper to handle assertion statement failures using a catch (AssertionException ae) block. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Option A is correct because it is sometimes advisable to thrown an assertion error even if assertions have been disabled.  Option B is incorrect because it is considered appropriate to check argument values in private methods using assertions.  Option C is incorrect; finally is never bypassed.  Option D is incorrect because AssertionErrors should never be handled.  Learn more problems on : [Assertions](http://www.indiabix.com/java-programming/assertions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/assertions/discussion-220) |

|  |  |
| --- | --- |
| 18. | What will be the output of the program?  public class WrapTest  {  public static void main(String [] args)  {  int result = 0;  short s = 42;  Long x = new Long("42");  Long y = new Long(42);  Short z = new Short("42");  Short x2 = new Short(s);  Integer y2 = new Integer("42");  Integer z2 = new Integer(42);  if (x == y) /\* Line 13 \*/  result = 1;  if (x.equals(y) ) /\* Line 15 \*/  result = result + 10;  if (x.equals(z) ) /\* Line 17 \*/  result = result + 100;  if (x.equals(x2) ) /\* Line 19 \*/  result = result + 1000;  if (x.equals(z2) ) /\* Line 21 \*/  result = result + 10000;  System.out.println("result = " + result);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | result = 1 | | |  | B. | |  | | --- | | result = 10 | | |  | C. | |  | | --- | | result = 11 | | |  | D. | |  | | --- | | result = 11010 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Line 13 fails because == compares reference values, not object values. Line 15 succeeds because both String and primitive wrapper constructors resolve to the same value (except for the Character wrapper). Lines 17, 19, and 21 fail because the equals() method fails if the object classes being compared are different and not in the same tree hierarchy.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-241) |

|  |  |
| --- | --- |
| 19. | What will be the output of the program?  class Q207  {  public static void main(String[] args)  {  int i1 = 5;  int i2 = 6;  String s1 = "7";  System.out.println(i1 + i2 + s1); /\* Line 8 \*/  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 18 | | |  | B. | |  | | --- | | 117 | | |  | C. | |  | | --- | | 567 | | |  | D. | |  | | --- | | Compiler error | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  This question is about the + (plus) operator and the overriden + (string cocatanation) operator. The rules that apply when you have a mixed expression of numbers and strings are:  If either operand is a String, the + operator concatenates the operands.  If both operands are numeric, the + operator adds the operands.  The expression on line 6 above can be read as "Add the values i1 and i2 together, then take the sum and convert it to a string and concatenate it with the String from the variable s1". In code, the compiler probably interprets the expression on line 8 above as:  System.out.println( new StringBuffer()  .append(new Integer(i1 + i2).toString())  .append(s1)  .toString() );  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-257) |

|  |  |
| --- | --- |
| 20. | What will be the output of the program?  public class StringRef  {  public static void main(String [] args)  {  String s1 = "abc";  String s2 = "def";  String s3 = s2; /\* Line 7 \*/  s2 = "ghi";  System.out.println(s1 + s2 + s3);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | abcdefghi | | |  | B. | |  | | --- | | abcdefdef | | |  | C. | |  | | --- | | abcghidef | | |  | D. | |  | | --- | | abcghighi | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  After line 7 executes, both s2 and s3 refer to a String object that contains the value "def". When line 8 executes, a new String object is created with the value "ghi", to which s2 refers. The reference variable s3still refers to the (immutable) String object with the value "def".  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-231) |

# Java Programming Test 7

[Home](http://www.indiabix.com/) » [Online Test](http://www.indiabix.com/online-test/categories/) » [Online Java Programming Test](http://www.indiabix.com/online-test/java-programming-test/) » Java Programming Test 7

|  |  |  |
| --- | --- | --- |
| **Marks : 0/20** | | |
| Total number of questions | : | **20** |
| Number of answered questions | : | **0** |
| Number of unanswered questions | : | **20** |

### Test Review : View answers and explanation for this test.

|  |  |
| --- | --- |
| 1. | Which cause a compiler error? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | int[ ] scores = {3, 5, 7}; | | |  | B. | |  | | --- | | int [ ][ ] scores = {2,7,6}, {9,3,45}; | | |  | C. | |  | | --- | | String cats[ ] = {"Fluffy", "Spot", "Zeus"}; | | |  | D. | |  | | --- | | boolean results[ ] = new boolean [] {true, false, true}; | | |  | E. | |  | | --- | | Integer results[ ] = {new Integer(3), new Integer(5), new Integer(8)}; | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B generates a compiler error: <identifier> expected. The compiler thinks you are trying to create two arrays because there are two array initialisers to the right of the equals, whereas your intention was to create one 3 x 3 two-dimensional array.  To correct the problem and make option B compile you need to add an extra pair of curly brackets:  int [ ] [ ] scores = { {2,7,6}, {9,3,45} };  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-33) |

|  |  |
| --- | --- |
| 2. | Which three are valid method signatures in an interface?   1. private int getArea(); 2. public float getVol(float x); 3. public void main(String [] args); 4. public static void main(String [] args); 5. boolean setFlag(Boolean [] test); |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 2 | | |  | B. | |  | | --- | | 2, 3 and 5 | | |  | C. | |  | | --- | | 3, 4, and 5 | | |  | D. | |  | | --- | | 2 and 4 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  (2), (3), and (5). These are all valid interface method signatures.  (1), is incorrect because an interface method must be public; if it is not explicitly declared public it will be made public implicitly. (4) is incorrect because interface methods cannot be static.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-30) |

|  |  |
| --- | --- |
| 3. | What will be the output of the program?  class Super  {  public Integer getLength()  {  return new Integer(4);  }  }  public class Sub extends Super  {  public Long getLength()  {  return new Long(5);  }  public static void main(String[] args)  {  Super sooper = new Super();  Sub sub = new Sub();  System.out.println(  sooper.getLength().toString() + "," + sub.getLength().toString() );  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 4, 4 | | |  | B. | |  | | --- | | 4, 5 | | |  | C. | |  | | --- | | 5, 4 | | |  | D. | |  | | --- | | Compilation fails. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Option D is correct, compilation fails - The return type of getLength( ) in the super class is an object of reference type Integer and the return type in the sub class is an object of reference type Long. In other words, it is not an override because of the change in the return type and it is also not an overload because the argument list has not changed.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-49) |

|  |  |
| --- | --- |
| 4. | What will be the output of the program?  class Test  {  public static void main(String [] args)  {  int x= 0;  int y= 0;  for (int z = 0; z < 5; z++)  {  if (( ++x > 2 ) && (++y > 2))  {  x++;  }  }  System.out.println(x + " " + y);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 5 2 | | |  | B. | |  | | --- | | 5 3 | | |  | C. | |  | | --- | | 6 3 | | |  | D. | |  | | --- | | 6 4 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  In the first two iterations x is incremented once and y is not because of the short circuit && operator. In the third and forth iterations x and y are each incremented, and in the fifth iteration x is doubly incremented and yis incremented.  Learn more problems on : [Operators and Assignments](http://www.indiabix.com/java-programming/operators-and-assignments/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/operators-and-assignments/discussion-62) |

|  |  |
| --- | --- |
| 5. | What will be the output of the program?  int i = l, j = -1;  switch (i)  {  case 0, 1: j = 1; /\* Line 4 \*/  case 2: j = 2;  default: j = 0;  }  System.out.println("j = " + j); |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | j = -1 | | |  | B. | |  | | --- | | j = 0 | | |  | C. | |  | | --- | | j = 1 | | |  | D. | |  | | --- | | Compilation fails. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  The case statement takes only a single argument. The case statement on line 4 is given two arguments so the compiler complains.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-97) |

|  |  |
| --- | --- |
| 6. | What will be the output of the program?  for(int i = 0; i < 3; i++)  {  switch(i)  {  case 0: break;  case 1: System.out.print("one ");  case 2: System.out.print("two ");  case 3: System.out.print("three ");  }  }  System.out.println("done"); |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | done | | |  | B. | |  | | --- | | one two done | | |  | C. | |  | | --- | | one two three done | | |  | D. | |  | | --- | | one two three two three done | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  The variable i will have the values 0, 1 and 2.  When i is 0, nothing will be printed because of the break in case 0.  When i is 1, "one two three" will be output because case 1, case 2 and case 3 will be executed (they don't have break statements).  When i is 2, "two three" will be output because case 2 and case 3 will be executed (again no break statements).  Finally, when the for loop finishes "done" will be output.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-95) |

|  |  |
| --- | --- |
| 7. | What will be the output of the program?  boolean bool = true;  if(bool = false) /\* Line 2 \*/  {  System.out.println("a");  }  else if(bool) /\* Line 6 \*/  {  System.out.println("b");  }  else if(!bool) /\* Line 10 \*/  {  System.out.println("c"); /\* Line 12 \*/  }  else  {  System.out.println("d");  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | a | | |  | B. | |  | | --- | | b | | |  | C. | |  | | --- | | c | | |  | D. | |  | | --- | | d | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Look closely at line 2, is this an equality check (==) or an assignment (=). The condition at line 2 evaluates to false and also assigns false to bool. bool is now false so the condition at line 6 is not true. The condition at line 10 checks to see if bool is not true ( if !(bool == true) ), it isn't so line 12 is executed.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-96) |

|  |  |
| --- | --- |
| 8. | What will be the output of the program?  class Exc0 extends Exception { }  class Exc1 extends Exc0 { } /\* Line 2 \*/  public class Test  {  public static void main(String args[])  {  try  {  throw new Exc1(); /\* Line 9 \*/  }  catch (Exc0 e0) /\* Line 11 \*/  {  System.out.println("Ex0 caught");  }  catch (Exception e)  {  System.out.println("exception caught");  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Ex0 caught | | |  | B. | |  | | --- | | exception caught | | |  | C. | |  | | --- | | Compilation fails because of an error at line 2. | | |  | D. | |  | | --- | | Compilation fails because of an error at line 9. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  An exception Exc1 is thrown and is caught by the catch statement on line 11. The code is executed in this block. There is no finally block of code to execute.  Learn more problems on : [Exceptions](http://www.indiabix.com/java-programming/exceptions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/exceptions/discussion-113) |

|  |  |
| --- | --- |
| 9. | public class MyProgram  {  public static void throwit()  {  throw new RuntimeException();  }  public static void main(String args[])  {  try  {  System.out.println("Hello world ");  throwit();  System.out.println("Done with try block ");  }  finally  {  System.out.println("Finally executing ");  }  }  }  which answer most closely indicates the behavior of the program? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | The program will not compile. | | |  | B. | |  | | --- | | The program will print Hello world, then will print that a RuntimeException has occurred, then will print Done with try block, and then will print Finally executing. | | |  | C. | |  | | --- | | The program will print Hello world, then will print that a RuntimeException has occurred, and then will print Finally executing. | | |  | D. | |  | | --- | | The program will print Hello world, then will print Finally executing, then will print that a RuntimeException has occurred. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Once the program throws a RuntimeException (in the throwit() method) that is not caught, the finally block will be executed and the program will be terminated. If a method does not handle an exception, the finally block is executed before the exception is propagated.  Learn more problems on : [Exceptions](http://www.indiabix.com/java-programming/exceptions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/exceptions/discussion-116) |

|  |  |
| --- | --- |
| 10. | Which interface provides the capability to store objects using a key-value pair? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Java.util.Map | | |  | B. | |  | | --- | | Java.util.Set | | |  | C. | |  | | --- | | Java.util.List | | |  | D. | |  | | --- | | Java.util.Collection | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  An object that maps keys to values. A map cannot contain duplicate keys; each key can map to at most one value.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-131) |

|  |  |
| --- | --- |
| 11. | Which of the following statements about the hashcode() method are incorrect?   1. The value returned by hashcode() is used in some collection classes to help locate objects. 2. The hashcode() method is required to return a positive int value. 3. The hashcode() method in the String class is the one inherited from Object. 4. Two new empty String objects will produce identical hashcodes. |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 2 | | |  | B. | |  | | --- | | 2 and 3 | | |  | C. | |  | | --- | | 3 and 4 | | |  | D. | |  | | --- | | 1 and 4 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  (2) is an incorrect statement because there is no such requirement.  (3) is an incorrect statement and therefore a correct answer because the hashcode for a string is computed from the characters in the string.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-146) |

|  |  |
| --- | --- |
| 12. | What will be the output of the program?  public class TestObj  {  public static void main (String [] args)  {  Object o = new Object() /\* Line 5 \*/  {  public boolean equals(Object obj)  {  return true;  }  } /\* Line 11 \*/    System.out.println(o.equals("Fred"));  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | It prints "true". | | |  | B. | |  | | --- | | It prints "Fred". | | |  | C. | |  | | --- | | An exception occurs at runtime. | | |  | D. | |  | | --- | | Compilation fails | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  This code would be legal if line 11 ended with a semicolon. Remember that line 5 is a statement that doesn't end until line 11, and a statement needs a closing semicolon!  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-158) |

|  |  |
| --- | --- |
| 13. | Which of the following will directly stop the execution of a Thread? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | wait() | | |  | B. | |  | | --- | | notify() | | |  | C. | |  | | --- | | notifyall() | | |  | D. | |  | | --- | | exits synchronized code | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Option A is correct. wait() causes the current thread to wait until another thread invokes the notify()method or the notifyAll() method for this object.  Option B is wrong. notify() - wakes up a single thread that is waiting on this object's monitor.  Option C is wrong. notifyAll() - wakes up all threads that are waiting on this object's monitor.  Option D is wrong. Typically, releasing a lock means the thread holding the lock (in other words, the thread currently in the synchronized method) exits the synchronized method. At that point, the lock is free until some other thread enters a synchronized method on that object. Does entering/exiting synchronized code mean that the thread execution stops? Not necessarily because the thread can still run code that is not synchronized. I think the word directly in the question gives us a clue. Exiting synchronized code does not directly stop the execution of a thread.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-164) |

|  |  |
| --- | --- |
| 14. | Which class or interface defines the wait(), notify(),and notifyAll() methods? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Object | | |  | B. | |  | | --- | | Thread | | |  | C. | |  | | --- | | Runnable | | |  | D. | |  | | --- | | Class | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  The Object class defines these thread-specific methods.  Option B, C, and D are incorrect because they do not define these methods. And yes, the Java API does define a class called Class, though you do not need to know it for the exam.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-167) |

|  |  |
| --- | --- |
| 15. | What will be the output of the program?  public class ThreadDemo  {  private int count = 1;  public synchronized void doSomething()  {  for (int i = 0; i < 10; i++)  System.out.println(count++);  }  public static void main(String[] args)  {  ThreadDemo demo = new ThreadDemo();  Thread a1 = new A(demo);  Thread a2 = new A(demo);  a1.start();  a2.start();  }  }  class A extends Thread  {  ThreadDemo demo;  public A(ThreadDemo td)  {  demo = td;  }  public void run()  {  demo.doSomething();  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | It will print the numbers 0 to 19 sequentially | | |  | B. | |  | | --- | | It will print the numbers 1 to 20 sequentially | | |  | C. | |  | | --- | | It will print the numbers 1 to 20, but the order cannot be determined | | |  | D. | |  | | --- | | The code will not compile. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  You have two different threads that share one reference to a common object.  The updating and output takes place inside synchronized code.  One thread will run to completion printing the numbers 1-10.  The second thread will then run to completion printing the numbers 11-20.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-192) |

|  |  |
| --- | --- |
| 16. | Which two can be used to create a new Thread?   1. Extend java.lang.Thread and override the run() method. 2. Extend java.lang.Runnable and override the start() method. 3. Implement java.lang.Thread and implement the run() method. 4. Implement java.lang.Runnable and implement the run() method. 5. Implement java.lang.Thread and implement the start() method. |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 2 | | |  | B. | |  | | --- | | 2 and 3 | | |  | C. | |  | | --- | | 1 and 4 | | |  | D. | |  | | --- | | 3 and 4 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  There are two ways of creating a thread; extend (sub-class) the Thread class and implement the Runnableinterface. For both of these ways you must implement (override and not overload) the public void run()method.  (1) is correct - Extending the Thread class and overriding its run method is a valid procedure.  (4) is correct - You must implement interfaces, and runnable is an interface and you must also include the run method.  (2) is wrong - Runnable is an interface which implements not Extends. Gives the error: (No interface expected here)  (3) is wrong - You cannot implement java.lang.Thread (This is a Class). (Implements Thread, gives the error: Interface expected). Implements expects an interface.  (5) is wrong - You cannot implement java.lang.Thread (This is a class). You Extend classes, and Implement interfaces. (Implements Thread, gives the error: Interface expected)  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-200) |

|  |  |
| --- | --- |
| 17. | The following block of code creates a Thread using a Runnable target:  Runnable target = new MyRunnable();  Thread myThread = new Thread(target);  Which of the following classes can be used to create the target, so that the preceding code compiles correctly? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | public class MyRunnable extends Runnable{public void run(){}} | | |  | B. | |  | | --- | | public class MyRunnable extends Object{public void run(){}} | | |  | C. | |  | | --- | | public class MyRunnable implements Runnable{public void run(){}} | | |  | D. | |  | | --- | | public class MyRunnable implements Runnable{void run(){}} | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  The class correctly implements the Runnable interface with a legal public void run() method.  Option A is incorrect because interfaces are not extended; they are implemented.  Option B is incorrect because even though the class would compile and it has a valid public void run()method, it does not implement the Runnable interface, so the compiler would complain when creating a Thread with an instance of it.  Option D is incorrect because the run() method must be public.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-201) |

|  |  |
| --- | --- |
| 18. | public class Myfile  {  public static void main (String[] args)  {  String biz = args[1];  String baz = args[2];  String rip = args[3];  System.out.println("Arg is " + rip);  }  }  Select how you would start the program to cause it to print: Arg is 2 |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | java Myfile 222 | | |  | B. | |  | | --- | | java Myfile 1 2 2 3 4 | | |  | C. | |  | | --- | | java Myfile 1 3 2 2 | | |  | D. | |  | | --- | | java Myfile 0 1 2 3 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Arguments start at array element 0 so the fourth arguement must be 2 to produce the correct output.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-230) |

|  |  |
| --- | --- |
| 19. | What will be the output of the program?  int i = (int) Math.random(); |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | i = 0 | | |  | B. | |  | | --- | | i = 1 | | |  | C. | |  | | --- | | value of i is undetermined | | |  | D. | |  | | --- | | Statement causes a compile error | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Math.random() returns a double value greater than or equal to 0 and less than 1. Its value is stored to an int but as this is a narrowing conversion, a cast is needed to tell the compiler that you are aware that there may be a loss of precision.  The value after the decimal point is lost when you cast a double to int and you are left with 0.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-258) |

|  |  |
| --- | --- |
| 20. | What will be the output of the program?  class A  {  public A(int x){}  }  class B extends A { }  public class test  {  public static void main (String args [])  {  A a = new B();  System.out.println("complete");  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | It compiles and runs printing nothing | | |  | B. | |  | | --- | | Compiles but fails at runtime | | |  | C. | |  | | --- | | Compile Error | | |  | D. | |  | | --- | | Prints "complete" | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  No constructor has been defined for class B therefore it will make a call to the default constructor but since class B extends class A it will also call the Super() default constructor.  Since a constructor has been defined in class A java will no longer supply a default constructor for class Atherefore when class B calls class A's default constructor it will result in a compile error.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-256) |

# Java Programming Test 6

[Home](http://www.indiabix.com/) » [Online Test](http://www.indiabix.com/online-test/categories/) » [Online Java Programming Test](http://www.indiabix.com/online-test/java-programming-test/) » Java Programming Test 6

|  |  |  |
| --- | --- | --- |
| **Marks : 0/20** | | |
| Total number of questions | : | **20** |
| Number of answered questions | : | **0** |
| Number of unanswered questions | : | **20** |

### Test Review : View answers and explanation for this test.

|  |  |
| --- | --- |
| 1. | What is the most restrictive access modifier that will allow members of one class to have access to members of another class in the same package? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | public | | |  | B. | |  | | --- | | abstract | | |  | C. | |  | | --- | | protected | | |  | D. | |  | | --- | | synchronized | | |  | E. | |  | | --- | | default access | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **E**  Explanation:  default access is the "package oriented" access modifier.  Option A and C are wrong because public and protected are less restrictive. Option B and D are wrong because abstract and synchronized are not access modifiers.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-24) |

|  |  |
| --- | --- |
| 2. | You want a class to have access to members of another class in the same package. Which is the most restrictive access that accomplishes this objective? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | public | | |  | B. | |  | | --- | | private | | |  | C. | |  | | --- | | protected | | |  | D. | |  | | --- | | default access | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  The only two real contenders are C and D. Protected access Option C makes a member accessible only to classes in the same package or subclass of the class. While default access Option D makes a member accessible only to classes in the same package.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-37) |

|  |  |
| --- | --- |
| 3. | Which is a valid declaration within an interface? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | public static short stop = 23; | | |  | B. | |  | | --- | | protected short stop = 23; | | |  | C. | |  | | --- | | transient short stop = 23; | | |  | D. | |  | | --- | | final void madness(short stop); | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  (A) is valid interface declarations.  (B) and (C) are incorrect because interface variables cannot be either protected or transient. (D) is incorrect because interface methods cannot be final or static.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-28) |

|  |  |
| --- | --- |
| 4. | switch(x)  {  default:  System.out.println("Hello");  }  Which two are acceptable types for x?   1. byte 2. long 3. char 4. float 5. Short 6. Long |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 3 | | |  | B. | |  | | --- | | 2 and 4 | | |  | C. | |  | | --- | | 3 and 5 | | |  | D. | |  | | --- | | 4 and 6 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Switch statements are based on integer expressions and since both bytes and chars can implicitly be widened to an integer, these can also be used. Also shorts can be used. Short and Long are wrapper classes and reference types can not be used as variables.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-79) |

|  |  |
| --- | --- |
| 5. | What will be the output of the program?  public class Switch2  {  final static short x = 2;  public static int y = 0;  public static void main(String [] args)  {  for (int z=0; z < 3; z++)  {  switch (z)  {  case x: System.out.print("0 ");  case x-1: System.out.print("1 ");  case x-2: System.out.print("2 ");  }  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 0 1 2 | | |  | B. | |  | | --- | | 0 1 2 1 2 2 | | |  | C. | |  | | --- | | 2 1 0 1 0 0 | | |  | D. | |  | | --- | | 2 1 2 0 1 2 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  The case expressions are all legal because x is marked final, which means the expressions can be evaluated at compile time. In the first iteration of the for loop case x-2 matches, so 2 is printed. In the second iteration, x-1 is matched so 1 and 2 are printed (remember, once a match is found all remaining statements are executed until a break statement is encountered). In the third iteration, x is matched. So 0 1 and 2 are printed.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-84) |

|  |  |
| --- | --- |
| 6. | What will be the output of the program?  public class Switch2  {  final static short x = 2;  public static int y = 0;  public static void main(String [] args)  {  for (int z=0; z < 3; z++)  {  switch (z)  {  case y: System.out.print("0 "); /\* Line 11 \*/  case x-1: System.out.print("1 "); /\* Line 12 \*/  case x: System.out.print("2 "); /\* Line 13 \*/  }  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 0 1 2 | | |  | B. | |  | | --- | | 0 1 2 1 2 2 | | |  | C. | |  | | --- | | Compilation fails at line 11. | | |  | D. | |  | | --- | | Compilation fails at line 12. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Case expressions must be constant expressions. Since x is marked final, lines 12 and 13 are legal; however y is not a final so the compiler will fail at line 11.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-83) |

|  |  |
| --- | --- |
| 7. | What will be the output of the program?  public class Switch2  {  final static short x = 2;  public static int y = 0;  public static void main(String [] args)  {  for (int z=0; z < 4; z++)  {  switch (z)  {  case x: System.out.print("0 ");  default: System.out.print("def ");  case x-1: System.out.print("1 ");  break;  case x-2: System.out.print("2 ");  }  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 0 def 1 | | |  | B. | |  | | --- | | 2 1 0 def 1 | | |  | C. | |  | | --- | | 2 1 0 def def | | |  | D. | |  | | --- | | 2 1 0 def 1 def 1 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  When z == 0 , case x-2 is matched. When z == 1, case x-1 is matched and then the break occurs. When z == 2, case x, then default, then x-1 are all matched. When z == 3, default, then x-1 are matched. The rules for default are that it will fall through from above like any other case (for instance when z == 2), and that it will match when no other cases match (for instance when z==3).  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-86) |

|  |  |
| --- | --- |
| 8. | Which collection class allows you to grow or shrink its size and provides indexed access to its elements, but whose methods are not synchronized? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | java.util.HashSet | | |  | B. | |  | | --- | | java.util.LinkedHashSet | | |  | C. | |  | | --- | | java.util.List | | |  | D. | |  | | --- | | java.util.ArrayList | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  All of the collection classes allow you to grow or shrink the size of your collection. ArrayList provides an index to its elements. The newer collection classes tend not to have synchronized methods. Vector is an older implementation of ArrayList functionality and has synchronized methods; it is slower than ArrayList.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-122) |

|  |  |
| --- | --- |
| 9. | What will be the output of the program?  public class Test  {  public static void main (String args[])  {  String str = NULL;  System.out.println(str);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | NULL | | |  | B. | |  | | --- | | Compile Error | | |  | C. | |  | | --- | | Code runs but no output | | |  | D. | |  | | --- | | Runtime Exception | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B is correct because to set the value of a String variable to null you must use "null" and not "NULL".  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-141) |

|  |  |
| --- | --- |
| 10. | Which of the following will not directly cause a thread to stop? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | notify() | | |  | B. | |  | | --- | | wait() | | |  | C. | |  | | --- | | InputStream access | | |  | D. | |  | | --- | | sleep() | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Option A is correct. notify() - wakes up a single thread that is waiting on this object's monitor.  Option B is wrong. wait() causes the current thread to wait until another thread invokes the notify()method or the notifyAll() method for this object.  Option C is wrong. Methods of the InputStream class block until input data is available, the end of the stream is detected, or an exception is thrown. Blocking means that a thread may stop until certain conditions are met.  Option D is wrong. sleep() - Causes the currently executing thread to sleep (temporarily cease execution) for a specified number of milliseconds. The thread does not lose ownership of any monitors.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-165) |

|  |  |
| --- | --- |
| 11. | What will be the output of the program?  public class SyncTest  {  public static void main (String [] args)  {  Thread t = new Thread()  {  Foo f = new Foo();  public void run()  {  f.increase(20);  }  };  t.start();  }  }  class Foo  {  private int data = 23;  public void increase(int amt)  {  int x = data;  data = x + amt;  }  }  and assuming that data must be protected from corruption, whatâ€”if anythingâ€”can you add to the preceding code to ensure the integrity of data? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Synchronize the run method. | | |  | B. | |  | | --- | | Wrap a synchronize(this) around the call to f.increase(). | | |  | C. | |  | | --- | | The existing code will cause a runtime exception. | | |  | D. | |  | | --- | | Synchronize the increase() method | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Option D is correct because synchronizing the code that actually does the increase will protect the code from being accessed by more than one thread at a time.  Option A is incorrect because synchronizing the run() method would stop other threads from running the run() method (a bad idea) but still would not prevent other threads with other runnables from accessing the increase() method.  Option B is incorrect for virtually the same reason as Aâ€”synchronizing the code that calls the increase()method does not prevent other code from calling the increase() method.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-184) |

|  |  |
| --- | --- |
| 12. | What will be the output of the program?  class Test116  {  static final StringBuffer sb1 = new StringBuffer();  static final StringBuffer sb2 = new StringBuffer();  public static void main(String args[])  {  new Thread()  {  public void run()  {  synchronized(sb1)  {  sb1.append("A");  sb2.append("B");  }  }  }.start();  new Thread()  {  public void run()  {  synchronized(sb1)  {  sb1.append("C");  sb2.append("D");  }  }  }.start(); /\* Line 28 \*/  System.out.println (sb1 + " " + sb2);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | main() will finish before starting threads. | | |  | B. | |  | | --- | | main() will finish in the middle of one thread. | | |  | C. | |  | | --- | | main() will finish after one thread. | | |  | D. | |  | | --- | | Cannot be determined. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Can you guarantee the order in which threads are going to run? No you can't. So how do you know what the output will be? The output cannot be determined.  add this code after line 28:  try { Thread.sleep(5000); } catch(InterruptedException e) { }  and you have some chance of predicting the outcome.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-187) |

|  |  |
| --- | --- |
| 13. | Which statement is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | If only one thread is blocked in the wait method of an object, and another thread executes the modify on that same object, then the first thread immediately resumes execution. | | |  | B. | |  | | --- | | If a thread is blocked in the wait method of an object, and another thread executes the notify method on the same object, it is still possible that the first thread might never resume execution. | | |  | C. | |  | | --- | | If a thread is blocked in the wait method of an object, and another thread executes the notify method on the same object, then the first thread definitely resumes execution as a direct and sole consequence of the notify call. | | |  | D. | |  | | --- | | If two threads are blocked in the wait method of one object, and another thread executes the notify method on the same object, then the first thread that executed the wait call first definitely resumes execution as a direct and sole consequence of the notify call. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B is correct - The notify method only wakes the thread. It does not guarantee that the thread will run.  Option A is incorrect - just because another thread activates the modify method in A this does not mean that the thread will automatically resume execution  Option C is incorrect - This is incorrect because as said in Answer B notify only wakes the thread but further to this once it is awake it goes back into the stack and awaits execution therefore it is not a "direct and sole consequence of the notify call"  Option D is incorrect - The notify method wakes one waiting thread up. If there are more than one sleeping threads then the choice as to which thread to wake is made by the machine rather than you therefore you cannot guarantee that the notify'ed thread will be the first waiting thread.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-198) |

|  |  |
| --- | --- |
| 14. | Which statement is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Memory is reclaimed by calling Runtime.gc(). | | |  | B. | |  | | --- | | Objects are not collected if they are accessible from live threads. | | |  | C. | |  | | --- | | An OutOfMemory error is only thrown if a single block of memory cannot be found that is large enough for a particular requirement. | | |  | D. | |  | | --- | | Objects that have finalize() methods always have their finalize() methods called before the program ends. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B is correct. If an object can be accessed from a live thread, it can't be garbage collected.  Option A is wrong. Runtime.gc() asks the garbage collector to run, but the garbage collector never makes any guarantees about when it will run or what unreachable objects it will free from memory.  Option C is wrong. The garbage collector runs immediately the system is out of memory before an OutOfMemoryException is thrown by the JVM.  Option D is wrong. If this were the case then the garbage collector would actively hang onto objects until a program finishes - this goes against the purpose of the garbage collector.  Learn more problems on : [Garbage Collections](http://www.indiabix.com/java-programming/garbage-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/garbage-collections/discussion-212) |

|  |  |
| --- | --- |
| 15. | What will be the output of the program (when you run with the -ea option) ?  public class Test  {  public static void main(String[] args)  {  int x = 0;  assert (x > 0) : "assertion failed"; /\* Line 6 \*/  System.out.println("finished");  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | finished | | |  | B. | |  | | --- | | Compilation fails. | | |  | C. | |  | | --- | | An AssertionError is thrown. | | |  | D. | |  | | --- | | An AssertionError is thrown and finished is output. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  An assertion Error is thrown as normal giving the output "assertion failed". The word "finished" is not printed (ensure you run with the -ea option)  Assertion failures are generally labeled in the stack trace with the file and line number from which they were thrown, and also in this case with the error's detail message "assertion failed". The detail message is supplied by the assert statement in line 6.  Learn more problems on : [Assertions](http://www.indiabix.com/java-programming/assertions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/assertions/discussion-215) |

|  |  |
| --- | --- |
| 16. | public class Test2  {  public static int x;  public static int foo(int y)  {  return y \* 2;  }  public static void main(String [] args)  {  int z = 5;  assert z > 0; /\* Line 11 \*/  assert z > 2: foo(z); /\* Line 12 \*/  if ( z < 7 )  assert z > 4; /\* Line 14 \*/  switch (z)  {  case 4: System.out.println("4 ");  case 5: System.out.println("5 ");  default: assert z < 10;  }  if ( z < 10 )  assert z > 4: z++; /\* Line 22 \*/  System.out.println(z);  }  }  which line is an example of an inappropriate use of assertions? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Line 11 | | |  | B. | |  | | --- | | Line 12 | | |  | C. | |  | | --- | | Line 14 | | |  | D. | |  | | --- | | Line 22 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Assert statements should not cause side effects. Line 22 changes the value of z if the assert statement is false.  Option A is fine; a second expression in an assert statement is not required.  Option B is fine because it is perfectly acceptable to call a method with the second expression of an assert statement.  Option C is fine because it is proper to call an assert statement conditionally.  Learn more problems on : [Assertions](http://www.indiabix.com/java-programming/assertions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/assertions/discussion-218) |

|  |  |
| --- | --- |
| 17. | Which statement is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Assertions can be enabled or disabled on a class-by-class basis. | | |  | B. | |  | | --- | | Conditional compilation is used to allow tested classes to run at full speed. | | |  | C. | |  | | --- | | Assertions are appropriate for checking the validity of arguments in a method. | | |  | D. | |  | | --- | | The programmer can choose to execute a return statement or to throw an exception if an assertion fails. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Option A is correct. The assertion status can be set for a named top-level class and any nested classes contained therein. This setting takes precedence over the class loader's default assertion status, and over any applicable per-package default. If the named class is not a top-level class, the change of status will have no effect on the actual assertion status of any class.  Option B is wrong. Is there such a thing as conditional compilation in Java?  Option C is wrong. For private methods - yes. But do not use assertions to check the parameters of a public method. An assert is inappropriate in public methods because the method guarantees that it will always enforce the argument checks. A public method must check its arguments whether or not assertions are enabled. Further, the assert construct does not throw an exception of the specified type. It can throw only an AssertionError.  Option D is wrong. Because you're never supposed to handle an assertion failure. That means don't catch it with a catch clause and attempt to recover.  Learn more problems on : [Assertions](http://www.indiabix.com/java-programming/assertions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/assertions/discussion-224) |

|  |  |
| --- | --- |
| 18. | What will be the output of the program?  String x = "xyz";  x.toUpperCase(); /\* Line 2 \*/  String y = x.replace('Y', 'y');  y = y + "abc";  System.out.println(y); |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | abcXyZ | | |  | B. | |  | | --- | | abcxyz | | |  | C. | |  | | --- | | xyzabc | | |  | D. | |  | | --- | | XyZabc | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Line 2 creates a new String object with the value "XYZ", but this new object is immediately lost because there is no reference to it. Line 3 creates a new String object referenced by y. This new String object has the value "xyz" because there was no "Y" in the String object referred to by x. Line 4 creates a new Stringobject, appends "abc" to the value "xyz", and refers y to the result.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-232) |

|  |  |
| --- | --- |
| 19. | What will be the output of the program?  class Tree { }  class Pine extends Tree { }  class Oak extends Tree { }  public class Forest1  {  public static void main (String [] args)  {  Tree tree = new Pine();  if( tree instanceof Pine )  System.out.println ("Pine");  else if( tree instanceof Tree )  System.out.println ("Tree");  else if( tree instanceof Oak )  System.out.println ( "Oak" );  else  System.out.println ("Oops ");  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Pine | | |  | B. | |  | | --- | | Tree | | |  | C. | |  | | --- | | Forest | | |  | D. | |  | | --- | | Oops | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  The program prints "Pine".  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-247) |

|  |  |
| --- | --- |
| 20. | What two statements are true about the result obtained from calling Math.random()?   1. The result is less than 0.0. 2. The result is greater than or equal to 0.0.. 3. The result is less than 1.0. 4. The result is greater than 1.0. 5. The result is greater than or equal to 1.0. |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 2 | | |  | B. | |  | | --- | | 2 and 3 | | |  | C. | |  | | --- | | 3 and 4 | | |  | D. | |  | | --- | | 4 and 5 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  (2) and (3) are correct. The result range for random() is 0.0 to < 1.0; 1.0 is not in range.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-259) |

# Java Programming Test 5

[Home](http://www.indiabix.com/) » [Online Test](http://www.indiabix.com/online-test/categories/) » [Online Java Programming Test](http://www.indiabix.com/online-test/java-programming-test/) » Java Programming Test 5

|  |  |  |
| --- | --- | --- |
| **Marks : 0/20** | | |
| Total number of questions | : | **20** |
| Number of answered questions | : | **0** |
| Number of unanswered questions | : | **20** |

### Test Review : View answers and explanation for this test.

|  |  |
| --- | --- |
| 1. | Given a method in a protected class, what access modifier do you use to restrict access to that method to only the other members of the same class? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | final | | |  | B. | |  | | --- | | static | | |  | C. | |  | | --- | | private | | |  | D. | |  | | --- | | protected | | |  | E. | |  | | --- | | volatile | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  The private access modifier limits access to members of the same class.  Option A, B, D, and E are wrong because protected are the wrong access modifiers, and final, static, and volatile are modifiers but not access modifiers.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-25) |

|  |  |
| --- | --- |
| 2. | interface DoMath  {  double getArea(int rad);  }  interface MathPlus  {  double getVol(int b, int h);  }  /\* Missing Statements ? \*/  which two code fragments inserted at end of the program, will allow to compile?   1. class AllMath extends DoMath { double getArea(int r); } 2. interface AllMath implements MathPlus { double getVol(int x, int y); } 3. interface AllMath extends DoMath { float getAvg(int h, int l); } 4. class AllMath implements MathPlus { double getArea(int rad); } 5. abstract class AllMath implements DoMath, MathPlus { public double getArea(int rad) { return rad \* rad \* 3.14; } } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 only | | |  | B. | |  | | --- | | 2 only | | |  | C. | |  | | --- | | 3 and 5 | | |  | D. | |  | | --- | | 1 and 4 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  (3) are (5) are correct because interfaces and abstract classes do not need to fully implement the interfaces they extend or implement (respectively).  (1) is incorrect because a class cannot extend an interface. (2) is incorrect because an interface cannot implement anything. (4) is incorrect because the method being implemented is from the wrong interface.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-56) |

|  |  |
| --- | --- |
| 3. | Which three statements are true?   1. The default constructor initialises method variables. 2. The default constructor has the same access as its class. 3. The default constructor invokes the no-arg constructor of the superclass. 4. If a class lacks a no-arg constructor, the compiler always creates a default constructor. 5. The compiler creates a default constructor only when there are no other constructors for the class. |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1, 2 and 4 | | |  | B. | |  | | --- | | 2, 3 and 5 | | |  | C. | |  | | --- | | 3, 4 and 5 | | |  | D. | |  | | --- | | 1, 2 and 3 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  (2) sounds correct as in the example below  class CoffeeCup {  private int innerCoffee;  public CoffeeCup() {  }    public void add(int amount) {  innerCoffee += amount;  }  //...  }  The compiler gives default constructors the same access level as their class. In the example above, class CoffeeCup is public, so the default constructor is public. If CoffeeCup had been given package access, the default constructor would be given package access as well.  (3) is correct. The Java compiler generates at least one instance initialisation method for every class it compiles. In the Java class file, the instance initialisation method is named "<init>." For each constructor in the source code of a class, the Java compiler generates one <init>() method. If the class declares no constructors explicitly, the compiler generates a default no-arg constructor that just invokes the superclass's no-arg constructor. As with any other constructor, the compiler creates an <init>() method in the class file that corresponds to this default constructor.  (5) is correct. The compiler creates a default constructor if you do not declare any constructors in your class.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-54) |

|  |  |
| --- | --- |
| 4. | What will be the output of the program?  class Test  {  public static void main(String [] args)  {  int x=20;  String sup = (x < 15) ? "small" : (x < 22)? "tiny" : "huge";  System.out.println(sup);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | small | | |  | B. | |  | | --- | | tiny | | |  | C. | |  | | --- | | huge | | |  | D. | |  | | --- | | Compilation fails | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  This is an example of a nested ternary operator. The second evaluation (x < 22) is true, so the "tiny" value is assigned to sup.  Learn more problems on : [Operators and Assignments](http://www.indiabix.com/java-programming/operators-and-assignments/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/operators-and-assignments/discussion-64) |

|  |  |
| --- | --- |
| 5. | Which of the following are legal lines of code?   1. int w = (int)888.8; 2. byte x = (byte)1000L; 3. long y = (byte)100; 4. byte z = (byte)100L; |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 2 | | |  | B. | |  | | --- | | 2 and 3 | | |  | C. | |  | | --- | | 3 and 4 | | |  | D. | |  | | --- | | All statements are correct. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Statements (1), (2), (3), and (4) are correct. (1) is correct because when a floating-point number (a double in this case) is cast to an int, it simply loses the digits after the decimal.  (2) and (4) are correct because a long can be cast into a byte. If the long is over 127, it loses its most significant (leftmost) bits.  (3) actually works, even though a cast is not necessary, because a long can store a byte.  Learn more problems on : [Operators and Assignments](http://www.indiabix.com/java-programming/operators-and-assignments/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/operators-and-assignments/discussion-74) |

|  |  |
| --- | --- |
| 6. | Which two statements are equivalent?   1. 16\*4 2. 16>>2 3. 16/2^2 4. 16>>>2 |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 2 | | |  | B. | |  | | --- | | 2 and 4 | | |  | C. | |  | | --- | | 3 and 4 | | |  | D. | |  | | --- | | 1 and 3 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  (2) is correct. 16 >> 2 = 4  (4) is correct. 16 >>> 2 = 4  (1) is wrong. 16 \* 4 = 64  (3) is wrong. 16/2 ^ 2 = 10  Learn more problems on : [Operators and Assignments](http://www.indiabix.com/java-programming/operators-and-assignments/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/operators-and-assignments/discussion-77) |

|  |  |
| --- | --- |
| 7. | What will be the output of the program?  public class Test  {  public static void main(String [] args)  {  int I = 1;  do while ( I < 1 )  System.out.print("I is " + I);  while ( I > 1 ) ;  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | I is 1 | | |  | B. | |  | | --- | | I is 1 I is 1 | | |  | C. | |  | | --- | | No output is produced. | | |  | D. | |  | | --- | | Compilation error | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  There are two different looping constructs in this problem. The first is a do-while loop and the second is a while loop, nested inside the do-while. The body of the do-while is only a single statement-brackets are not needed. You are assured that the while expression will be evaluated at least once, followed by an evaluation of the do-while expression. Both expressions are false and no output is produced.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-88) |

|  |  |
| --- | --- |
| 8. | What will be the output of the program?  public class RTExcept  {  public static void throwit ()  {  System.out.print("throwit ");  throw new RuntimeException();  }  public static void main(String [] args)  {  try  {  System.out.print("hello ");  throwit();  }  catch (Exception re )  {  System.out.print("caught ");  }  finally  {  System.out.print("finally ");  }  System.out.println("after ");  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | hello throwit caught | | |  | B. | |  | | --- | | Compilation fails | | |  | C. | |  | | --- | | hello throwit RuntimeException caught after | | |  | D. | |  | | --- | | hello throwit caught finally after | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  The main() method properly catches and handles the RuntimeException in the catch block, finally runs (as it always does), and then the code returns to normal.  A, B and C are incorrect based on the program logic described above. Remember that properly handled exceptions do not cause the program to stop executing.  Learn more problems on : [Exceptions](http://www.indiabix.com/java-programming/exceptions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/exceptions/discussion-105) |

|  |  |
| --- | --- |
| 9. | What will be the output of the program?  public class X  {  public static void main(String [] args)  {  try  {  badMethod();  System.out.print("A");  }  catch (Exception ex)  {  System.out.print("B");  }  finally  {  System.out.print("C");  }  System.out.print("D");  }  public static void badMethod() {}  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | AC | | |  | B. | |  | | --- | | BC | | |  | C. | |  | | --- | | ACD | | |  | D. | |  | | --- | | ABCD | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  There is no exception thrown, so all the code with the exception of the catch statement block is run.  Learn more problems on : [Exceptions](http://www.indiabix.com/java-programming/exceptions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/exceptions/discussion-111) |

|  |  |
| --- | --- |
| 10. | Which of the following are Java reserved words?   1. run 2. import 3. default 4. implement |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 2 | | |  | B. | |  | | --- | | 2 and 3 | | |  | C. | |  | | --- | | 3 and 4 | | |  | D. | |  | | --- | | 2 and 4 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  (2) - This is a Java keyword  (3) - This is a Java keyword  (1) - Is incorrect because although it is a method of Thread/Runnable it is not a keyword  (4) - This is not a Java keyword the keyword is implements  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-126) |

|  |  |
| --- | --- |
| 11. | What will be the output of the program?  public class Test  {  public static void main (String[] args)  {  String foo = args[1];  String bar = args[2];  String baz = args[3];  System.out.println("baz = " + baz); /\* Line 8 \*/  }  }  And the command line invocation:  > java Test red green blue |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | baz = | | |  | B. | |  | | --- | | baz = null | | |  | C. | |  | | --- | | baz = blue | | |  | D. | |  | | --- | | Runtime Exception | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  When running the program you entered 3 arguments "red", "green" and "blue". When dealing with arrays in java you must remember ALL ARRAYS IN JAVA ARE ZERO BASED therefore args[0] becomes "red", args[1] becomes "green" and args[2] becomes "blue".  When the program entcounters line 8 above at runtime it looks for args[3] which has never been created therefore you get an  ArrayIndexOutOfBoundsException at runtime.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-138) |

|  |  |
| --- | --- |
| 12. | What will be the output of the program?  class Happy extends Thread  {  final StringBuffer sb1 = new StringBuffer();  final StringBuffer sb2 = new StringBuffer();  public static void main(String args[])  {  final Happy h = new Happy();  new Thread()  {  public void run()  {  synchronized(this)  {  h.sb1.append("A");  h.sb2.append("B");  System.out.println(h.sb1);  System.out.println(h.sb2);  }  }  }.start();  new Thread()  {  public void run()  {  synchronized(this)  {  h.sb1.append("D");  h.sb2.append("C");  System.out.println(h.sb2);  System.out.println(h.sb1);  }  }  }.start();  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | ABBCAD | | |  | B. | |  | | --- | | ABCBCAD | | |  | C. | |  | | --- | | CDADACB | | |  | D. | |  | | --- | | Output determined by the underlying platform. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Can you guarantee the order in which threads are going to run? No you can't. So how do you know what the output will be? The output cannot be determined.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-188) |

|  |  |
| --- | --- |
| 13. | What will be the output of the program?  class MyThread extends Thread  {  public static void main(String [] args)  {  MyThread t = new MyThread();  Thread x = new Thread(t);  x.start(); /\* Line 7 \*/  }  public void run()  {  for(int i = 0; i < 3; ++i)  {  System.out.print(i + "..");  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Compilation fails. | | |  | B. | |  | | --- | | 1..2..3.. | | |  | C. | |  | | --- | | 0..1..2..3.. | | |  | D. | |  | | --- | | 0..1..2.. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  The thread MyThread will start and loop three times (from 0 to 2).  Option A is incorrect because the Thread class implements the Runnable interface; therefore, in line 7, Thread can take an object of type Thread as an argument in the constructor.  Option B and C are incorrect because the variable i in the for loop starts with a value of 0 and ends with a value of 2.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-180) |

|  |  |
| --- | --- |
| 14. | Which statement is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | A static method cannot be synchronized. | | |  | B. | |  | | --- | | If a class has synchronized code, multiple threads can still access the nonsynchronized code. | | |  | C. | |  | | --- | | Variables can be protected from concurrent access problems by marking them with the synchronized keyword. | | |  | D. | |  | | --- | | When a thread sleeps, it releases its locks. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  B is correct because multiple threads are allowed to enter nonsynchronized code, even within a class that has some synchronized methods.  A is incorrect because static methods can be synchronized; they synchronize on the lock on the instance of class java.lang.Class that represents the class type.  C is incorrect because only methodsâ€”not variablesâ€”can be marked synchronized.  D is incorrect because a sleeping thread still maintains its locks.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-196) |

|  |  |
| --- | --- |
| 15. | Which statement is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Calling Runtime.gc() will cause eligible objects to be garbage collected. | | |  | B. | |  | | --- | | The garbage collector uses a mark and sweep algorithm. | | |  | C. | |  | | --- | | If an object can be accessed from a live thread, it can't be garbage collected. | | |  | D. | |  | | --- | | If object 1 refers to object 2, then object 2 can't be garbage collected. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  This is a great way to think about when objects can be garbage collected.  Option A and B assume guarantees that the garbage collector never makes.  Option D is wrong because of the now famous islands of isolation scenario.  Learn more problems on : [Garbage Collections](http://www.indiabix.com/java-programming/garbage-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/garbage-collections/discussion-211) |

|  |  |
| --- | --- |
| 16. | Which statement is true about assertions in the Java programming language? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Assertion expressions should not contain side effects. | | |  | B. | |  | | --- | | Assertion expression values can be any primitive type. | | |  | C. | |  | | --- | | Assertions should be used for enforcing preconditions on public methods. | | |  | D. | |  | | --- | | An AssertionError thrown as a result of a failed assertion should always be handled by the enclosing method. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Option A is correct. Because assertions may be disabled, programs must not assume that the boolean expressions contained in assertions will be evaluated. Thus these expressions should be free of side effects. That is, evaluating such an expression should not affect any state that is visible after the evaluation is complete. Although it is not illegal for a boolean expression contained in an assertion to have a side effect, it is generally inappropriate, as it could cause program behaviour to vary depending on whether assertions are enabled or disabled.  Assertion checking may be disabled for increased performance. Typically, assertion checking is enabled during program development and testing and disabled for deployment.  Option B is wrong. Because you assert that something is "true". True is Boolean. So, an expression must evaluate to Boolean, not int or byte or anything else. Use the same rules for an assertion expression that you would use for a while condition.  Option C is wrong. Usually, enforcing a precondition on a public method is done by condition-checking code that you write yourself, to give you specific exceptions.  Option D is wrong. "You're never supposed to handle an assertion failure"  Not all legal uses of assertions are considered appropriate. As with so much of Java, you can abuse the intended use for assertions, despite the best efforts of Sun's Java engineers to discourage you. For example, you're never supposed to handle an assertion failure. That means don't catch it with a catch clause and attempt to recover. Legally, however, AssertionError is a subclass of Throwable, so it can be caught. But just don't do it! If you're going to try to recover from something, it should be an exception. To discourage you from trying to substitute an assertion for an exception, the AssertionError doesn't provide access to the object that generated it. All you get is the String message.  Learn more problems on : [Assertions](http://www.indiabix.com/java-programming/assertions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/assertions/discussion-223) |

|  |  |
| --- | --- |
| 17. | What will be the output of the program?  public class ObjComp  {  public static void main(String [] args )  {  int result = 0;  ObjComp oc = new ObjComp();  Object o = oc;  if (o == oc)  result = 1;  if (o != oc)  result = result + 10;  if (o.equals(oc) )  result = result + 100;  if (oc.equals(o) )  result = result + 1000;  System.out.println("result = " + result);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 | | |  | B. | |  | | --- | | 10 | | |  | C. | |  | | --- | | 101 | | |  | D. | |  | | --- | | 1101 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Even though o and oc are reference variables of different types, they are both referring to the same object. This means that == will resolve to true and that the default equals() method will also resolve to true.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-243) |

|  |  |
| --- | --- |
| 18. | What will be the output of the program?  public class Test178  {  public static void main(String[] args)  {  String s = "foo";  Object o = (Object)s;  if (s.equals(o))  {  System.out.print("AAA");  }  else  {  System.out.print("BBB");  }  if (o.equals(s))  {  System.out.print("CCC");  }  else  {  System.out.print("DDD");  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | AAACCC | | |  | B. | |  | | --- | | AAADDD | | |  | C. | |  | | --- | | BBBCCC | | |  | D. | |  | | --- | | BBBDDD | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-250) |

|  |  |
| --- | --- |
| 19. | What will be the output of the program?  int i = 1, j = 10;  do  {  if(i++ > --j) /\* Line 4 \*/  {  continue;  }  } while (i < 5);  System.out.println("i = " + i + "and j = " + j); /\* Line 9 \*/ |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | i = 6 and j = 5 | | |  | B. | |  | | --- | | i = 5 and j = 5 | | |  | C. | |  | | --- | | i = 6 and j = 6 | | |  | D. | |  | | --- | | i = 5 and j = 6 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  This question is not testing your knowledge of the continue statement. It is testing your knowledge of the order of evaluation of operands. Basically the prefix and postfix unary operators have a higher order of evaluation than the relational operators. So on line 4 the variable i is incremented and the variable j is decremented before the greater than comparison is made. As the loop executes the comparison on line 4 will be:  if(i > j)  if(2 > 9)  if(3 > 8)  if(4 > 7)  if(5 > 6) at this point i is not less than 5, therefore the loop terminates and line 9 outputs the values of i and jas 5 and 6 respectively.  The continue statement never gets to execute because i never reaches a value that is greater than j.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-249) |

|  |  |
| --- | --- |
| 20. | What will be the output of the program?  public class ExamQuestion7  {  static int j;  static void methodA(int i)  {  boolean b;  do  {  b = i<10 | methodB(4); /\* Line 9 \*/  b = i<10 || methodB(8); /\* Line 10 \*/  }while (!b);  }  static boolean methodB(int i)  {  j += i;  return true;  }  public static void main(String[] args)  {  methodA(0);  System.out.println( "j = " + j );  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | j = 0 | | |  | B. | |  | | --- | | j = 4 | | |  | C. | |  | | --- | | j = 8 | | |  | D. | |  | | --- | | The code will run with no output | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  The lines to watch here are lines 9 & 10. Line 9 features the non-shortcut version of the OR operator so both of its operands will be evaluated and therefore methodB(4) is executed.  However line 10 has the shortcut version of the OR operator and if the 1st of its operands evaluates to true (which in this case is true), then the 2nd operand isn't evaluated, so methodB(8) never gets called.  The loop is only executed once, b is initialized to false and is assigned true on line 9. Thus j = 4.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-253) |

**Java Programming Test 4**

|  |
| --- |
|  |
|

|  |  |
| --- | --- |
| 3. | class A  {  protected int method1(int a, int b)  {  return 0;  }  }  Which is valid in a class that extends class A? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | public int method1(int a, int b) {return 0; } | | |  | B. | |  | | --- | | private int method1(int a, int b) { return 0; } | | |  | C. | |  | | --- | | public short method1(int a, int b) { return 0; } | | |  | D. | |  | | --- | | static protected int method1(int a, int b) { return 0; } | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Option A is correct - because the class that extends A is just simply overriding method1.  Option B is wrong - because it can't override as there are less access privileges in the subclass method1.  Option C is wrong - because to override it, the return type needs to be an integer. The different return type means that the method is not overriding but the same argument list means that the method is not overloading. Conflict - compile time error.  Option D is wrong - because you can't override a method and make it a class method i.e. using static.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-41) |

|  |  |
| --- | --- |
| 4. | What will be the output of the program?  class Test  {  public static void main(String [] args)  {  Test p = new Test();  p.start();  }  void start()  {  boolean b1 = false;  boolean b2 = fix(b1);  System.out.println(b1 + " " + b2);  }  boolean fix(boolean b1)  {  b1 = true;  return b1;  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | true true | | |  | B. | |  | | --- | | false true | | |  | C. | |  | | --- | | true false | | |  | D. | |  | | --- | | false false | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  The boolean b1 in the fix() method is a different boolean than the b1 in the start() method. The b1 in the start() method is not updated by the fix() method.  Learn more problems on : [Operators and Assignments](http://www.indiabix.com/java-programming/operators-and-assignments/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/operators-and-assignments/discussion-67) |

|  |
| --- |
| . |
|

|  |
| --- |
|  |
|

|  |  |
| --- | --- |
| 10. | What will be the output of the program?  public class Test  {  public static void main(String args[])  {  int i = 1, j = 0;  switch(i)  {  case 2: j += 6;  case 4: j += 1;  default: j += 2;  case 0: j += 4;  }  System.out.println("j = " + j);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | j = 0 | | |  | B. | |  | | --- | | j = 2 | | |  | C. | |  | | --- | | j = 4 | | |  | D. | |  | | --- | | j = 6 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Because there are no break statements, the program gets to the default case and adds 2 to j, then goes to case 0 and adds 4 to the new j. The result is j = 6.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-93) |

|  |  |
| --- | --- |
| 11. | import java.io.\*;  public class MyProgram  {  public static void main(String args[])  {  FileOutputStream out = null;  try  {  out = new FileOutputStream("test.txt");  out.write(122);  }  catch(IOException io)  {  System.out.println("IO Error.");  }  finally  {  out.close();  }  }  }  and given that all methods of class FileOutputStream, including close(), throw an IOException, which of these is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | This program will compile successfully. | | |  | B. | |  | | --- | | This program fails to compile due to an error at line 4. | | |  | C. | |  | | --- | | This program fails to compile due to an error at line 6. | | |  | D. | |  | | --- | | This program fails to compile due to an error at line 18. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Any method (in this case, the main() method) that throws a checked exception (in this case, out.close() ) must be called within a try clause, or the method must declare that it throws the exception. Either main()must declare that it throws an exception, or the call to out.close() in the finally block must fall inside a (in this case nested) try-catch block.  Learn more problems on : [Exceptions](http://www.indiabix.com/java-programming/exceptions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/exceptions/discussion-115) |

|  |  |
| --- | --- |
| 12. | Which statement is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | catch(X x) can catch subclasses of X where X is a subclass of Exception. | | |  | B. | |  | | --- | | The Error class is a RuntimeException. | | |  | C. | |  | | --- | | Any statement that can throw an Error must be enclosed in a try block. | | |  | D. | |  | | --- | | Any statement that can throw an Exception must be enclosed in a try block. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Option A is correct. If the class specified in the catch clause does have subclasses, any exception object that subclasses the specified class will be caught as well.  Option B is wrong. The error class is a subclass of Throwable and not Runtime Exception.  Option C is wrong. You do not catch this class of error.  Option D is wrong. An exception can be thrown to the next method higher up the call stack.  Learn more problems on : [Exceptions](http://www.indiabix.com/java-programming/exceptions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/exceptions/discussion-118) |

|  |
| --- |
| 13. |
|

|  |  |
| --- | --- |
| 14. | What will be the output of the program?  import java.util.\*;  class H  {  public static void main (String[] args)  {  Object x = new Vector().elements();  System.out.print((x instanceof Enumeration)+",");  System.out.print((x instanceof Iterator)+",");  System.out.print(x instanceof ListIterator);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Prints: false,false,false | | |  | B. | |  | | --- | | Prints: false,false,true | | |  | C. | |  | | --- | | Prints: false,true,false | | |  | D. | |  | | --- | | Prints: true,false,false | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  The Vector.elements method returns an Enumeration over the elements of the vector. Vectorimplements the List interface and extends AbstractList so it is also possible to get an Iterator over a Vector by invoking the iterator or listIterator method.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-135) |

|  |  |
| --- | --- |
| 15. | What will be the output of the program?  TreeSet map = new TreeSet();  map.add("one");  map.add("two");  map.add("three");  map.add("four");  map.add("one");  Iterator it = map.iterator();  while (it.hasNext() )  {  System.out.print( it.next() + " " );  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | one two three four | | |  | B. | |  | | --- | | four three two one | | |  | C. | |  | | --- | | four one three two | | |  | D. | |  | | --- | | one two three four one | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  TreeSet assures no duplicate entries; also, when it is accessed it will return elements in natural order, which typically means alphabetical.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-142) |

|  |  |
| --- | --- |
| 16. | class Foo  {  class Bar{ }  }  class Test  {  public static void main (String [] args)  {  Foo f = new Foo();  /\* Line 10: Missing statement ? \*/  }  }  which statement, inserted at line 10, creates an instance of Bar? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Foo.Bar b = new Foo.Bar(); | | |  | B. | |  | | --- | | Foo.Bar b = f.new Bar(); | | |  | C. | |  | | --- | | Bar b = new f.Bar(); | | |  | D. | |  | | --- | | Bar b = f.new Bar(); | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B is correct because the syntax is correct-using both names (the enclosing class and the inner class) in the reference declaration, then using a reference to the enclosing class to invoke new on the inner class.  Option A, C and D all use incorrect syntax. A is incorrect because it doesn't use a reference to the enclosing class, and also because it includes both names in the new.  C is incorrect because it doesn't use the enclosing class name in the reference variable declaration, and because the new syntax is wrong.  D is incorrect because it doesn't use the enclosing class name in the reference variable declaration.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-155) |

|  |  |
| --- | --- |
| 17. | Which three guarantee that a thread will leave the running state?   1. yield() 2. wait() 3. notify() 4. notifyAll() 5. sleep(1000) 6. aLiveThread.join() 7. Thread.killThread() |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1, 2 and 4 | | |  | B. | |  | | --- | | 2, 5 and 6 | | |  | C. | |  | | --- | | 3, 4 and 7 | | |  | D. | |  | | --- | | 4, 5 and 7 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  (2) is correct because wait() always causes the current thread to go into the object's wait pool.  (5) is correct because sleep() will always pause the currently running thread for at least the duration specified in the sleep argument (unless an interrupted exception is thrown).  (6) is correct because, assuming that the thread you're calling join() on is alive, the thread calling join()will immediately block until the thread you're calling join() on is no longer alive.  (1) is wrong, but tempting. The yield() method is not guaranteed to cause a thread to leave the running state, although if there are runnable threads of the same priority as the currently running thread, then the current thread will probably leave the running state.  (3) and (4) are incorrect because they don't cause the thread invoking them to leave the running state.  (7) is wrong because there's no such method.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-176) |

|  |  |
| --- | --- |
| 19. | What will be the output of the program?  public class Test  {  public static void main(String[] args)  {  int x = 0;  assert (x > 0) ? "assertion failed" : "assertion passed" ;  System.out.println("finished");  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | finished | | |  | B. | |  | | --- | | Compiliation fails. | | |  | C. | |  | | --- | | An AssertionError is thrown and finished is output. | | |  | D. | |  | | --- | | An AssertionError is thrown with the message "assertion failed." | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Compilation Fails. You can't use the Assert statement in a similar way to the ternary operator. Don't confuse.  Learn more problems on : [Assertions](http://www.indiabix.com/java-programming/assertions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/assertions/discussion-216) |

**Java Programming Test 3**

|  |  |
| --- | --- |
| 1. | public class Test { }  What is the prototype of the default constructor? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Test( ) | | |  | B. | |  | | --- | | Test(void) | | |  | C. | |  | | --- | | public Test( ) | | |  | D. | |  | | --- | | public Test(void) | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Option A and B are wrong because they use the default access modifier and the access modifier for the class is public (remember, the default constructor has the same access modifier as the class).  Option D is wrong. The void makes the compiler think that this is a method specification - in fact if it were a method specification the compiler would spit it out. |

|  |  |
| --- | --- |
| 2. | What is the widest valid returnType for methodA in line 3?  public class ReturnIt  {  returnType methodA(byte x, double y) /\* Line 3 \*/  {  return (long)x / y \* 2;  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | int | | |  | B. | |  | | --- | | byte | | |  | C. | |  | | --- | | long | | |  | D. | |  | | --- | | double | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  However A, B and C are all wrong. Each of these would result in a narrowing conversion. Whereas we want a widening conversion, therefore the only correct answer is D. Don't be put off by the long cast, this applies only to the variable x and not the rest of the expression. It is the variable y (of type double) that forces the widening conversion to double.  Java's widening conversions are:  - From a byte to a short, an int, a long, a float, or a double.  - From a short, an int, a long, a float, or a double.  - From a char to an int, a long, a float, or a double.  - From an int to a long, a float, or a double.  - From a long to a float, or a double.  - From a float to a double.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-40) |

|  |  |
| --- | --- |
| 3. | Which two cause a compiler error?   1. float[ ] f = new float(3); 2. float f2[ ] = new float[ ]; 3. float[ ]f1 = new float[3]; 4. float f3[ ] = new float[3]; 5. float f5[ ] = {1.0f, 2.0f, 2.0f}; |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 2, 4 | | |  | B. | |  | | --- | | 3, 5 | | |  | C. | |  | | --- | | 4, 5 | | |  | D. | |  | | --- | | 1, 2 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  (1) causes two compiler errors ( '[' expected and illegal start of expression) because the wrong type of bracket is used, ( ) instead of [ ]. The following is the correct syntax: float[ ] f = new float[3];  (2) causes a compiler error ( '{' expected ) because the array constructor does not specify the number of elements in the array. The following is the correct syntax: float f2[ ] = new float[3];  (3), (4), and (5) compile without error.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-35) |

|  |  |
| --- | --- |
| 4. | What will be the output of the program?  public class Test  {  public static void main(String args[])  {  class Foo  {  public int i = 3;  }  Object o = (Object)new Foo();  Foo foo = (Foo)o;  System.out.println("i = " + foo.i);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | i = 3 | | |  | B. | |  | | --- | | Compilation fails. | | |  | C. | |  | | --- | | i = 5 | | |  | D. | |  | | --- | | A ClassCastException will occur. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-44) |

|  |
| --- |
| 5. |
|

|  |  |
| --- | --- |
| 6. | What will be the output of the program?  class Test  {  public static void main(String [] args)  {  int x= 0;  int y= 0;  for (int z = 0; z < 5; z++)  {  if (( ++x > 2 ) || (++y > 2))  {  x++;  }  }  System.out.println(x + " " + y);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 5 3 | | |  | B. | |  | | --- | | 8 2 | | |  | C. | |  | | --- | | 8 3 | | |  | D. | |  | | --- | | 8 5 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  The first two iterations of the for loop both x and y are incremented. On the third iteration x is incremented, and for the first time becomes greater than 2. The short circuit or operator || keeps y from ever being incremented again and x is incremented twice on each of the last three iterations.  Learn more problems on : [Operators and Assignments](http://www.indiabix.com/java-programming/operators-and-assignments/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/operators-and-assignments/discussion-61) |

|  |  |
| --- | --- |
|  |  |

|  |
| --- |
|  |
|

|  |
| --- |
| 9. |
|

|  |  |
| --- | --- |
| 10. | Which class does not override the equals() and hashCode() methods, inheriting them directly from class Object? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | java.lang.String | | |  | B. | |  | | --- | | java.lang.Double | | |  | C. | |  | | --- | | java.lang.StringBuffer | | |  | D. | |  | | --- | | java.lang.Character | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  java.lang.StringBuffer is the only class in the list that uses the default methods provided by class Object.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-121) |

|  |
| --- |
| 11. |
|

|  |  |
| --- | --- |
| 12. | /\* Missing Statement ? \*/  public class foo  {  public static void main(String[]args)throws Exception  {  java.io.PrintWriter out = new java.io.PrintWriter();  new java.io.OutputStreamWriter(System.out,true);  out.println("Hello");  }  }  What line of code should replace the missing statement to make this program compile? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | No statement required. | | |  | B. | |  | | --- | | import java.io.\*; | | |  | C. | |  | | --- | | include java.io.\*; | | |  | D. | |  | | --- | | import java.io.PrintWriter; | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  The usual method for using/importing the java packages/classes is by using an import statement at the top of your code. However it is possible to explicitly import the specific class that you want to use as you use it which is shown in the code above. The disadvantage of this however is that every time you create a new object you will have to use the class path in the case "java.io" then the class name in the long run leading to a lot more typing.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-133) |

|  |  |
| --- | --- |
| 13. | Which is true about an anonymous inner class? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | It can extend exactly one class and implement exactly one interface. | | |  | B. | |  | | --- | | It can extend exactly one class and can implement multiple interfaces. | | |  | C. | |  | | --- | | It can extend exactly one class or implement exactly one interface. | | |  | D. | |  | | --- | | It can implement multiple interfaces regardless of whether it also extends a class. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Option C is correct because the syntax of an anonymous inner class allows for only one named type after the new, and that type must be either a single interface (in which case the anonymous class implements that one interface) or a single class (in which case the anonymous class extends that one class).  Option A, B, D, and E are all incorrect because they don't follow the syntax rules described in the response for answer Option C.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-157) |

|  |  |
| --- | --- |
| 14. | public class MyOuter  {  public static class MyInner  {  public static void foo() { }  }  }  which statement, if placed in a class other than MyOuter or MyInner, instantiates an instance of the nested class? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | MyOuter.MyInner m = new MyOuter.MyInner(); | | |  | B. | |  | | --- | | MyOuter.MyInner mi = new MyInner(); | | |  | C. | |  | | --- | | MyOuter m = new MyOuter();  MyOuter.MyInner mi = m.new MyOuter.MyInner(); | | |  | D. | |  | | --- | | MyInner mi = new MyOuter.MyInner(); | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  MyInner is a static nested class, so it must be instantiated using the fully-scoped name of MyOuter.MyInner.  Option B is incorrect because it doesn't use the enclosing name in the new.  Option C is incorrect because it uses incorrect syntax. When you instantiate a nested class by invoking new on an instance of the enclosing class, you do not use the enclosing name. The difference between Option A and C is that Option C is calling new on an instance of the enclosing class rather than just new by itself.  Option D is incorrect because it doesn't use the enclosing class name in the variable declaration.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-151) |

|  |  |
| --- | --- |
| 15. | Which two are valid constructors for Thread?   1. Thread(Runnable r, String name) 2. Thread() 3. Thread(int priority) 4. Thread(Runnable r, ThreadGroup g) 5. Thread(Runnable r, int priority) |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 3 | | |  | B. | |  | | --- | | 2 and 4 | | |  | C. | |  | | --- | | 1 and 2 | | |  | D. | |  | | --- | | 2 and 5 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  (1) and (2) are both valid constructors for Thread.  (3), (4), and (5) are not legal Thread constructors, although (4) is close. If you reverse the arguments in (4), you'd have a valid constructor.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-171) |

|  |  |
| --- | --- |
| 16. | Which two of the following methods are defined in class Thread?   1. start() 2. wait() 3. notify() 4. run() 5. terminate() |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 4 | | |  | B. | |  | | --- | | 2 and 3 | | |  | C. | |  | | --- | | 3 and 4 | | |  | D. | |  | | --- | | 2 and 4 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  (1) and (4). Only start() and run() are defined by the Thread class.  (2) and (3) are incorrect because they are methods of the Object class. (5) is incorrect because there's no such method in any thread-related class.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-172) |

|  |  |
| --- | --- |
| 17. | Under which conditions will a currently executing thread stop?   1. When an interrupted exception occurs. 2. When a thread of higher priority is ready (becomes runnable). 3. When the thread creates a new thread. 4. When the stop() method is called. |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 3 | | |  | B. | |  | | --- | | 2 and 4 | | |  | C. | |  | | --- | | 1 and 4 | | |  | D. | |  | | --- | | 2 and 3 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  The statements (2) and (4) makes currently executing thread to stop.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-170) |

|  |  |
| --- | --- |
| 18. | Which statement is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | The notifyAll() method must be called from a synchronized context. | | |  | B. | |  | | --- | | To call wait(), an object must own the lock on the thread. | | |  | C. | |  | | --- | | The notify() method is defined in class java.lang.Thread. | | |  | D. | |  | | --- | | The notify() method causes a thread to immediately release its locks. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Option A is correct because the notifyAll() method (along with wait() and notify()) must always be called from within a synchronized context.  Option B is incorrect because to call wait(), the thread must own the lock on the object that wait() is being invoked on, not the other way around.  Option C is wrong because notify() is defined in java.lang.Object.  Option D is wrong because notify() will not cause a thread to release its locks. The thread can only release its locks by exiting the synchronized code.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-197) |

**Java Programming Test 1**

[Home](http://www.indiabix.com/) » [Online Test](http://www.indiabix.com/online-test/categories/) » [Online Java Programming Test](http://www.indiabix.com/online-test/java-programming-test/) » Java Programming Test 1

|  |  |  |
| --- | --- | --- |
| **Marks : 0/20** | | |
| Total number of questions | : | **20** |
| Number of answered questions | : | **0** |
| Number of unanswered questions | : | **20** |

**Test Review : View answers and explanation for this test.**

|  |  |
| --- | --- |
| 1. | What will be the output of the program?  class A  {  final public int GetResult(int a, int b) { return 0; }  }  class B extends A  {  public int GetResult(int a, int b) {return 1; }  }  public class Test  {  public static void main(String args[])  {  B b = new B();  System.out.println("x = " + b.GetResult(0, 1));  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | x = 0 | | |  | B. | |  | | --- | | x = 1 | | |  | C. | |  | | --- | | Compilation fails. | | |  | D. | |  | | --- | | An exception is thrown at runtime. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  The code doesn't compile because the method GetResult() in class A is final and so cannot be overridden.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-50) |

|  |  |
| --- | --- |
| 2. | What will be the output of the program?  class SC2  {  public static void main(String [] args)  {  SC2 s = new SC2();  s.start();  }  void start()  {  int a = 3;  int b = 4;  System.out.print(" " + 7 + 2 + " ");  System.out.print(a + b);  System.out.print(" " + a + b + " ");  System.out.print(foo() + a + b + " ");  System.out.println(a + b + foo());  }  String foo()  {  return "foo";  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 9 7 7 foo 7 7foo | | |  | B. | |  | | --- | | 72 34 34 foo34 34foo | | |  | C. | |  | | --- | | 9 7 7 foo34 34foo | | |  | D. | |  | | --- | | 72 7 34 foo34 7foo | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Because all of these expressions use the + operator, there is no precedence to worry about and all of the expressions will be evaluated from left to right. If either operand being evaluated is a String, the + operator will concatenate the two operands; if both operands are numeric, the + operator will add the two operands.  Learn more problems on : [Operators and Assignments](http://www.indiabix.com/java-programming/operators-and-assignments/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/operators-and-assignments/discussion-69) |

|  |  |
| --- | --- |
| 3. | What will be the output of the program?  class BoolArray  {  boolean [] b = new boolean[3];  int count = 0;  void set(boolean [] x, int i)  {  x[i] = true;  ++count;  }  public static void main(String [] args)  {  BoolArray ba = new BoolArray();  ba.set(ba.b, 0);  ba.set(ba.b, 2);  ba.test();  }  void test()  {  if ( b[0] && b[1] | b[2] )  count++;  if ( b[1] && b[(++count - 2)] )  count += 7;  System.out.println("count = " + count);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | count = 0 | | |  | B. | |  | | --- | | count = 2 | | |  | C. | |  | | --- | | count = 3 | | |  | D. | |  | | --- | | count = 4 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  The reference variables b and x both refer to the same boolean array. count is incremented for each call to the set() method, and once again when the first if test is true. Because of the && short circuit operator, count is not incremented during the second if test.  Learn more problems on : [Operators and Assignments](http://www.indiabix.com/java-programming/operators-and-assignments/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/operators-and-assignments/discussion-65) |

|  |  |
| --- | --- |
| 4. | Which two statements are equivalent?   1. 3/2 2. 3<2 3. 3\*4 4. 3<<2 |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 and 2 | | |  | B. | |  | | --- | | 2 and 3 | | |  | C. | |  | | --- | | 3 and 4 | | |  | D. | |  | | --- | | 1 and 4 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  (1) is wrong. 3/2 = 1 (integer arithmetic).  (2) is wrong. 3 < 2 = false.  (3) is correct. 3 \* 4 = 12.  (4) is correct. 3 <<2= 12. In binary 3 is 11, now shift the bits two places to the left and we get 1100 which is 12 in binary (3\*2\*2).  Learn more problems on : [Operators and Assignments](http://www.indiabix.com/java-programming/operators-and-assignments/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/operators-and-assignments/discussion-76) |

|  |  |
| --- | --- |
| 5. | public void foo( boolean a, boolean b)  {  if( a )  {  System.out.println("A"); /\* Line 5 \*/  }  else if(a && b) /\* Line 7 \*/  {  System.out.println( "A && B");  }  else /\* Line 11 \*/  {  if ( !b )  {  System.out.println( "notB") ;  }  else  {  System.out.println( "ELSE" ) ;  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | If a is true and b is true then the output is "A && B" | | |  | B. | |  | | --- | | If a is true and b is false then the output is "notB" | | |  | C. | |  | | --- | | If a is false and b is true then the output is "ELSE" | | |  | D. | |  | | --- | | If a is false and b is false then the output is "ELSE" | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Option C is correct. The output is "ELSE". Only when a is false do the output lines after 11 get some chance of executing.  Option A is wrong. The output is "A". When a is true, irrespective of the value of b, only the line 5 output will be executed. The condition at line 7 will never be evaluated (when a is true it will always be trapped by the line 12 condition) therefore the output will never be "A && B".  Option B is wrong. The output is "A". When a is true, irrespective of the value of b, only the line 5 output will be executed.  Option D is wrong. The output is "notB".  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-82) |

|  |  |
| --- | --- |
| 6. | What will be the output of the program?  Float f = new Float("12");  switch (f)  {  case 12: System.out.println("Twelve");  case 0: System.out.println("Zero");  default: System.out.println("Default");  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Zero | | |  | B. | |  | | --- | | Twelve | | |  | C. | |  | | --- | | Default | | |  | D. | |  | | --- | | Compilation fails | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  The switch statement can only be supported by integers or variables more "narrow" than an integer i.e. byte, char, short. Here a Float wrapper object is used and so the compilation fails.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-94) |

|  |  |
| --- | --- |
| 7. | What will be the output of the program?  public class Test  {  public static void aMethod() throws Exception  {  try /\* Line 5 \*/  {  throw new Exception(); /\* Line 7 \*/  }  finally /\* Line 9 \*/  {  System.out.print("finally "); /\* Line 11 \*/  }  }  public static void main(String args[])  {  try  {  aMethod();  }  catch (Exception e) /\* Line 20 \*/  {  System.out.print("exception ");  }  System.out.print("finished"); /\* Line 24 \*/  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | finally | | |  | B. | |  | | --- | | exception finished | | |  | C. | |  | | --- | | finally exception finished | | |  | D. | |  | | --- | | Compilation fails | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  This is what happens:  (1) The execution of the try block (line 5) completes abruptly because of the throw statement (line 7).  (2) The exception cannot be assigned to the parameter of any catch clause of the try statement therefore the finally block is executed (line 9) and "finally" is output (line 11).  (3) The finally block completes normally, and then the try statement completes abruptly because of the throw statement (line 7).  (4) The exception is propagated up the call stack and is caught by the catch in the main method (line 20). This prints "exception".  (5) Lastly program execution continues, because the exception has been caught, and "finished" is output (line 24).  Learn more problems on : [Exceptions](http://www.indiabix.com/java-programming/exceptions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/exceptions/discussion-108) |

|  |  |
| --- | --- |
| 8. | Which statement is true for the class java.util.ArrayList? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | The elements in the collection are ordered. | | |  | B. | |  | | --- | | The collection is guaranteed to be immutable. | | |  | C. | |  | | --- | | The elements in the collection are guaranteed to be unique. | | |  | D. | |  | | --- | | The elements in the collection are accessed using a unique key. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  Yes, always the elements in the collection are ordered.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-145) |

|  |  |
| --- | --- |
| 9. | Which is true about a method-local inner class? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | It must be marked final. | | |  | B. | |  | | --- | | It can be marked abstract. | | |  | C. | |  | | --- | | It can be marked public. | | |  | D. | |  | | --- | | It can be marked static. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Option B is correct because a method-local inner class can be abstract, although it means a subclass of the inner class must be created if the abstract class is to be used (so an abstract method-local inner class is probably not useful).  Option A is incorrect because a method-local inner class does not have to be declared final (although it is legal to do so).  C and D are incorrect because a method-local inner class cannot be made public (remember-you cannot mark any local variables as public), or static.  Learn more problems on : [Inner Classes](http://www.indiabix.com/java-programming/inner-classes/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/inner-classes/discussion-156) |

|  |  |
| --- | --- |
| 10. | class X implements Runnable  {  public static void main(String args[])  {  /\* Missing code? \*/  }  public void run() {}  }  Which of the following line of code is suitable to start a thread ? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Thread t = new Thread(X); | | |  | B. | |  | | --- | | Thread t = new Thread(X); t.start(); | | |  | C. | |  | | --- | | X run = new X(); Thread t = new Thread(run); t.start(); | | |  | D. | |  | | --- | | Thread t = new Thread(); x.run(); | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  Option C is suitable to start a thread.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-177) |

|  |  |
| --- | --- |
| 11. | What will be the output of the program?  class MyThread extends Thread  {  public static void main(String [] args)  {  MyThread t = new MyThread();  t.start();  System.out.print("one. ");  t.start();  System.out.print("two. ");  }  public void run()  {  System.out.print("Thread ");  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Compilation fails | | |  | B. | |  | | --- | | An exception occurs at runtime. | | |  | C. | |  | | --- | | It prints "Thread one. Thread two." | | |  | D. | |  | | --- | | The output cannot be determined. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  When the start() method is attempted a second time on a single Thread object, the method will throw an IllegalThreadStateException (you will not need to know this exception name for the exam). Even if the thread has finished running, it is still illegal to call start() again.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-179) |

|  |  |
| --- | --- |
| 12. | What will be the output of the program?  class MyThread extends Thread  {  MyThread() {}  MyThread(Runnable r) {super(r); }  public void run()  {  System.out.print("Inside Thread ");  }  }  class MyRunnable implements Runnable  {  public void run()  {  System.out.print(" Inside Runnable");  }  }  class Test  {  public static void main(String[] args)  {  new MyThread().start();  new MyThread(new MyRunnable()).start();  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Prints "Inside Thread Inside Thread" | | |  | B. | |  | | --- | | Prints "Inside Thread Inside Runnable" | | |  | C. | |  | | --- | | Does not compile | | |  | D. | |  | | --- | | Throws exception at runtime | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **A**  Explanation:  If a Runnable object is passed to the Thread constructor, then the run method of the Thread class will invoke the run method of the Runnable object.  In this case, however, the run method in the Thread class is overridden by the run method in MyThread class. Therefore the run() method in MyRunnable is never invoked.  Both times, the run() method in MyThread is invoked instead.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-193) |

|  |  |
| --- | --- |
| 13. | What will be the output of the program?  class s implements Runnable  {  int x, y;  public void run()  {  for(int i = 0; i < 1000; i++)  synchronized(this)  {  x = 12;  y = 12;  }  System.out.print(x + " " + y + " ");  }  public static void main(String args[])  {  s run = new s();  Thread t1 = new Thread(run);  Thread t2 = new Thread(run);  t1.start();  t2.start();  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | DeadLock | | |  | B. | |  | | --- | | It print 12 12 12 12 | | |  | C. | |  | | --- | | Compilation Error | | |  | D. | |  | | --- | | Cannot determine output. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  The program will execute without any problems and print 12 12 12 12.  Learn more problems on : [Threads](http://www.indiabix.com/java-programming/threads/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/threads/discussion-189) |

|  |  |
| --- | --- |
| 14. | public class Test  {  public void foo()  {  assert false; /\* Line 5 \*/  assert false; /\* Line 6 \*/  }  public void bar()  {  while(true)  {  assert false; /\* Line 12 \*/  }  assert false; /\* Line 14 \*/  }  }  What causes compilation to fail? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Line 5 | | |  | B. | |  | | --- | | Line 6 | | |  | C. | |  | | --- | | Line 12 | | |  | D. | |  | | --- | | Line 14 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Option D is correct. Compilation fails because of an unreachable statement at line 14. It is a compile-time error if a statement cannot be executed because it is unreachable. The question is now, why is line 20 unreachable? If it is because of the assert then surely line 6 would also be unreachable. The answer must be something other than assert.  Examine the following:  A while statement can complete normally if and only if at least one of the following is true:  - The while statement is reachable and the condition expression is not a constant expression with value true.  -There is a reachable break statement that exits the while statement.  The while statement at line 11 is infinite and there is no break statement therefore line 14 is unreachable. You can test this with the following code:  public class Test80  {  public void foo()  {  assert false;  assert false;  }  public void bar()  {  while(true)  {  assert false;  break;  }  assert false;  }  }  Learn more problems on : [Assertions](http://www.indiabix.com/java-programming/assertions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/assertions/discussion-217) |

|  |  |
| --- | --- |
| 15. | What will be the output of the program?  public class Test  {  public static int y;  public static void foo(int x)  {  System.out.print("foo ");  y = x;  }  public static int bar(int z)  {  System.out.print("bar ");  return y = z;  }  public static void main(String [] args )  {  int t = 0;  assert t > 0 : bar(7);  assert t > 1 : foo(8); /\* Line 18 \*/  System.out.println("done ");  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | bar | | |  | B. | |  | | --- | | bar done | | |  | C. | |  | | --- | | foo done | | |  | D. | |  | | --- | | Compilation fails | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  The foo() method returns void. It is a perfectly acceptable method, but because it returns void it cannot be used in an assert statement, so line 18 will not compile.  Learn more problems on : [Assertions](http://www.indiabix.com/java-programming/assertions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/assertions/discussion-214) |

|  |  |
| --- | --- |
| 16. | Which of the following statements is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | In an assert statement, the expression after the colon ( : ) can be any Java expression. | | |  | B. | |  | | --- | | If a switch block has no default, adding an assert default is considered appropriate. | | |  | C. | |  | | --- | | In an assert statement, if the expression after the colon ( : ) does not have a value, the assert's error message will be empty. | | |  | D. | |  | | --- | | It is appropriate to handle assertion failures using a catch clause. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  Adding an assertion statement to a switch statement that previously had no default case is considered an excellent use of the assert mechanism.  Option A is incorrect because only Java expressions that return a value can be used. For instance, a method that returns void is illegal.  Option C is incorrect because the expression after the colon must have a value.  Option D is incorrect because assertions throw errors and not exceptions, and assertion errors do cause program termination and should not be handled.  Learn more problems on : [Assertions](http://www.indiabix.com/java-programming/assertions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/assertions/discussion-219) |

|  |  |
| --- | --- |
| 17. | public class Test2  {  public static int x;  public static int foo(int y)  {  return y \* 2;  }  public static void main(String [] args)  {  int z = 5;  assert z > 0; /\* Line 11 \*/  assert z > 2: foo(z); /\* Line 12 \*/  if ( z < 7 )  assert z > 4; /\* Line 14 \*/  switch (z)  {  case 4: System.out.println("4 ");  case 5: System.out.println("5 ");  default: assert z < 10;  }  if ( z < 10 )  assert z > 4: z++; /\* Line 22 \*/  System.out.println(z);  }  }  which line is an example of an inappropriate use of assertions? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Line 11 | | |  | B. | |  | | --- | | Line 12 | | |  | C. | |  | | --- | | Line 14 | | |  | D. | |  | | --- | | Line 22 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Assert statements should not cause side effects. Line 22 changes the value of z if the assert statement is false.  Option A is fine; a second expression in an assert statement is not required.  Option B is fine because it is perfectly acceptable to call a method with the second expression of an assert statement.  Option C is fine because it is proper to call an assert statement conditionally.  Learn more problems on : [Assertions](http://www.indiabix.com/java-programming/assertions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/assertions/discussion-226) |

|  |  |
| --- | --- |
| 18. | What will be the output of the program?  public class NFE  {  public static void main(String [] args)  {  String s = "42";  try  {  s = s.concat(".5"); /\* Line 8 \*/  double d = Double.parseDouble(s);  s = Double.toString(d);  int x = (int) Math.ceil(Double.valueOf(s).doubleValue());  System.out.println(x);  }  catch (NumberFormatException e)  {  System.out.println("bad number");  }  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 42 | | |  | B. | |  | | --- | | 42.5 | | |  | C. | |  | | --- | | 43 | | |  | D. | |  | | --- | | bad number | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  All of this code is legal, and line 8 creates a new String with a value of "42.5". Lines 9 and 10 convert the String to a double and then back again. Line 11 is funâ€” Math.ceil()'s argument expression is evaluated first. We invoke the valueOf() method that returns an anonymous Double object (with a value of 42.5). Then the doubleValue() method is called (invoked on the newly created Double object), and returns a double primitive (there and back again), with a value of (you guessed it) 42.5. The ceil() method converts this to 43.0, which is cast to an int and assigned to x.  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-238) |

|  |  |
| --- | --- |
| 19. | What will be the output of the program?  public class Test138  {  public static void stringReplace (String text)  {  text = text.replace ('j' , 'c'); /\* Line 5 \*/  }  public static void bufferReplace (StringBuffer text)  {  text = text.append ("c"); /\* Line 9 \*/  }  public static void main (String args[])  {  String textString = new String ("java");  StringBuffer textBuffer = new StringBuffer ("java"); /\* Line 14 \*/  stringReplace(textString);  bufferReplace(textBuffer);  System.out.println (textString + textBuffer);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | java | | |  | B. | |  | | --- | | javac | | |  | C. | |  | | --- | | javajavac | | |  | D. | |  | | --- | | Compile error | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **C**  Explanation:  A string is immutable, it cannot be changed, that's the reason for the StringBuffer class. The stringReplace method does not change the string declared on line 14, so this remains set to "java".  Method parameters are always passed by value - a copy is passed into the method - if the copy changes, the original remains intact, line 5 changes the reference i.e. text points to a new String object, however this is lost when the method completes. The textBuffer is a StringBuffer so it can be changed.  This change is carried out on line 9, so "java" becomes "javac", the text reference on line 9 remains unchanged. This gives us the output of "javajavac"  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-246) |

|  |  |
| --- | --- |
| 20. | What will be the output of the program (in jdk1.6 or above)?  public class BoolTest  {  public static void main(String [] args)  {  Boolean b1 = new Boolean("false");  boolean b2;  b2 = b1.booleanValue();  if (!b2)  {  b2 = true;  System.out.print("x ");  }  if (b1 & b2) /\* Line 13 \*/  {  System.out.print("y ");  }  System.out.println("z");  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | z | | |  | B. | |  | | --- | | x z | | |  | C. | |  | | --- | | y z | | |  | D. | |  | | --- | | Compilation fails. | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Learn more problems on : [Java.lang Class](http://www.indiabix.com/java-programming/java-lang-class/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/java-lang-class/discussion-239) |

**Java Programming Test 2**

|  |  |
| --- | --- |
| 1. | Which of the following class level (nonlocal) variable declarations will not compile? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | protected int a; | | |  | B. | |  | | --- | | transient int b = 3; | | |  | C. | |  | | --- | | private synchronized int e; | | |  | D. | |  |  | | --- | --- | | volatile int d; | http://www.indiabix.com/_files/images/website/wrong.gif | |   Your Answer: Option **D**  Correct Answer: Option **C**  Explanation:  Option C will not compile; the synchronized modifier applies only to methods.  Option A and B will compile because protected and transient are legal variable modifiers. Option D will compile because volatile is a proper variable modifier.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-29) |

|  |  |
| --- | --- |
| 3. | class A  {  A( ) { }  }  class B extends A  { }  Which statement is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Class B'S constructor is public. | | |  | B. | |  | | --- | | Class B'S constructor has no arguments. | | |  | C. | |  |  | | --- | --- | | Class B'S constructor includes a call to this( ). | http://www.indiabix.com/_files/images/website/wrong.gif | | |  | D. | |  | | --- | | None of these. | |   Your Answer: Option **C**  Correct Answer: Option **B**  Explanation:  Option B is correct. Class B inherits Class A's constructor which has no arguments.  Option A is wrong. Class B inherits Class A's constructor which uses default access.  Option C is wrong. There is just no implicit call to this( ).  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-53) |

|  |  |
| --- | --- |
| 4. | /\* Missing statements ? \*/  public class NewTreeSet extends java.util.TreeSet  {  public static void main(String [] args)  {  java.util.TreeSet t = new java.util.TreeSet();  t.clear();  }  public void clear()  {  TreeMap m = new TreeMap();  m.clear();  }  }  which two statements, added independently at beginning of the program, allow the code to compile?   1. No statement is required 2. import java.util.\*; 3. import.java.util.Tree\*; 4. import java.util.TreeSet; 5. import java.util.TreeMap; |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 1 only | | |  | B. | |  | | --- | | 2 and 5 | | |  | C. | |  | | --- | | 3 and 4 | | |  | D. | |  |  | | --- | --- | | 3 and 5 | http://www.indiabix.com/_files/images/website/wrong.gif | |   Your Answer: Option **D**  Correct Answer: Option **B**  Explanation:  (2) and (5). TreeMap is the only class that must be imported. TreeSet does not need an import statement because it is described with a fully qualified name.  (1) is incorrect because TreeMap must be imported. (3) is incorrect syntax for an import statement. (4) is incorrect because it will not import TreeMap, which is required.  Learn more problems on : [Declarations and Access Control](http://www.indiabix.com/java-programming/declarations-and-access-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/declarations-and-access-control/discussion-55) |

|  |
| --- |
|  |
|

|  |  |
| --- | --- |
| 7. | What will be the output of the program?  public class If1  {  static boolean b;  public static void main(String [] args)  {  short hand = 42;  if ( hand < 50 && !b ) /\* Line 7 \*/  hand++;  if ( hand > 50 ); /\* Line 9 \*/  else if ( hand > 40 )  {  hand += 7;  hand++;  }  else  --hand;  System.out.println(hand);  }  } |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 41 | | |  | B. | |  | | --- | | 42 | | |  | C. | |  |  | | --- | --- | | 50 | http://www.indiabix.com/_files/images/website/wrong.gif | | |  | D. | |  | | --- | | 51 | |   Your Answer: Option **C**  Correct Answer: Option **D**  Explanation:  In Java, boolean instance variables are initialized to false, so the if test on line 7 is true and hand is incremented. Line 9 is legal syntax, a do nothing statement. The else-if is true so hand has 7 added to it and is then incremented.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-85) |

|  |  |
| --- | --- |
| 8. | What will be the output of the program?  int i = 0;  while(1)  {  if(i == 4)  {  break;  }  ++i;  }  System.out.println("i = " + i); |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | i = 0 | | |  | B. | |  | | --- | | i = 3 | | |  | C. | |  |  | | --- | --- | | i = 4 | http://www.indiabix.com/_files/images/website/wrong.gif | | |  | D. | |  | | --- | | Compilation fails. | |   Your Answer: Option **C**  Correct Answer: Option **D**  Explanation:  Compilation fails because the argument of the while loop, the condition, must be of primitive type boolean. In Java, 1 does not represent the true state of a boolean, rather it is seen as an integer.  Learn more problems on : [Flow Control](http://www.indiabix.com/java-programming/flow-control/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/flow-control/discussion-98) |

|  |  |
| --- | --- |
| 11. | What will be the output of the program?  try  {  int x = 0;  int y = 5 / x;  }  catch (Exception e)  {  System.out.println("Exception");  }  catch (ArithmeticException ae)  {  System.out.println(" Arithmetic Exception");  }  System.out.println("finished"); |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | finished | | |  | B. | |  |  | | --- | --- | | Exception | http://www.indiabix.com/_files/images/website/wrong.gif | | |  | C. | |  | | --- | | Compilation fails. | | |  | D. | |  | | --- | | Arithmetic Exception | |   Your Answer: Option **B**  Correct Answer: Option **C**  Explanation:  Compilation fails because ArithmeticException has already been caught. ArithmeticException is a subclass of java.lang.Exception, by time the ArithmeticException has been specified it has already been caught by the Exception class.  If ArithmeticException appears before Exception, then the file will compile. When catching exceptions the more specific exceptions must be listed before the more general (the subclasses must be caught before the superclasses).  Learn more problems on : [Exceptions](http://www.indiabix.com/java-programming/exceptions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/exceptions/discussion-107) |

|  |  |
| --- | --- |
| 12. | Which statement is true? |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | A. | |  |  | | --- | --- | | A try statement must have at least one corresponding catch block. | http://www.indiabix.com/_files/images/website/wrong.gif | | |  | B. | |  | | --- | | Multiple catch statements can catch the same class of exception more than once. | | |  | C. | |  | | --- | | An Error that might be thrown in a method must be declared as thrown by that method, or be handled within that method. | | |  | D. | |  | | --- | | Except in case of VM shutdown, if a try block starts to execute, a corresponding finally block will always start to execute. | |   Your Answer: Option **A**  Correct Answer: Option **D**  Explanation:  A is wrong. A try statement can exist without catch, but it must have a finally statement.  B is wrong. A try statement executes a block. If a value is thrown and the try statement has one or more catch clauses that can catch it, then control will be transferred to the first such catch clause. If that catchblock completes normally, then the try statement completes normally.  C is wrong. Exceptions of type Error and RuntimeException do not have to be caught, only checked exceptions (java.lang.Exception) have to be caught. However, speaking of Exceptions, Exceptions do not have to be handled in the same method as the throw statement. They can be passed to another method.  If you put a finally block after a try and its associated catch blocks, then once execution enters the tryblock, the code in that finally block will definitely be executed except in the following circumstances:   1. An exception arising in the finally block itself. 2. The death of the thread. 3. The use of System.exit() 4. Turning off the power to the CPU.   I suppose the last three could be classified as VM shutdown.  Learn more problems on : [Exceptions](http://www.indiabix.com/java-programming/exceptions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/exceptions/discussion-120) |

|  |  |
| --- | --- |
| 13. | class Test1  {  public int value;  public int hashCode() { return 42; }  }  class Test2  {  public int value;  public int hashcode() { return (int)(value^5); }  }  which statement is true? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | class Test1 will not compile. | | |  | B. | |  | | --- | | The Test1 hashCode() method is more efficient than the Test2 hashCode() method. | | |  | C. | |  | | --- | | The Test1 hashCode() method is less efficient than the Test2 hashCode() method. | | |  | D. | |  |  | | --- | --- | | class Test2 will not compile. | http://www.indiabix.com/_files/images/website/wrong.gif | |   Your Answer: Option **D**  Correct Answer: Option **C**  Explanation:  The so-called "hashing algorithm" implemented by class Test1 will always return the same value, 42, which is legal but which will place all of the hash table entries into a single bucket, the most inefficient setup possible.  Option A and D are incorrect because these classes are legal.  Option B is incorrect based on the logic described above.  Learn more problems on : [Objects and Collections](http://www.indiabix.com/java-programming/objects-and-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/objects-and-collections/discussion-149) |

|  |  |
| --- | --- |
| 15. | class Bar { }  class Test  {  Bar doBar()  {  Bar b = new Bar(); /\* Line 6 \*/  return b; /\* Line 7 \*/  }  public static void main (String args[])  {  Test t = new Test(); /\* Line 11 \*/  Bar newBar = t.doBar(); /\* Line 12 \*/  System.out.println("newBar");  newBar = new Bar(); /\* Line 14 \*/  System.out.println("finishing"); /\* Line 15 \*/  }  }  At what point is the Bar object, created on line 6, eligible for garbage collection? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | after line 12 | | |  | B. | |  |  | | --- | --- | | after line 14 | http://www.indiabix.com/_files/images/website/accept.png | | |  | C. | |  | | --- | | after line 7, when doBar() completes | | |  | D. | |  | | --- | | after line 15, when main() completes | |   Your Answer: Option **B**  Correct Answer: Option **B**  Explanation:  Option B is correct. All references to the Bar object created on line 6 are destroyed when a new reference to a new Bar object is assigned to the variable newBar on line 14. Therefore the Bar object, created on line 6, is eligible for garbage collection after line 14.  Option A is wrong. This actually protects the object from garbage collection.  Option C is wrong. Because the reference in the doBar() method is returned on line 7 and is stored in newBar on line 12. This preserver the object created on line 6.  Option D is wrong. Not applicable because the object is eligible for garbage collection after line 14.  Learn more problems on : [Garbage Collections](http://www.indiabix.com/java-programming/garbage-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/garbage-collections/discussion-208) |

|  |  |
| --- | --- |
| 16. | class Test  {  private Demo d;  void start()  {  d = new Demo();  this.takeDemo(d); /\* Line 7 \*/  } /\* Line 8 \*/  void takeDemo(Demo demo)  {  demo = null;  demo = new Demo();  }  }  When is the Demo object eligible for garbage collection? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | After line 7 | | |  | B. | |  | | --- | | After line 8 | | |  | C. | |  |  | | --- | --- | | After the start() method completes | http://www.indiabix.com/_files/images/website/wrong.gif | | |  | D. | |  | | --- | | When the instance running this code is made eligible for garbage collection. | |   Your Answer: Option **C**  Correct Answer: Option **D**  Explanation:  Option D is correct. By a process of elimination.  Option A is wrong. The variable d is a member of the Test class and is never directly set to null.  Option B is wrong. A copy of the variable d is set to null and not the actual variable d.  Option C is wrong. The variable d exists outside the start() method (it is a class member). So, when the start() method finishes the variable d still holds a reference.  Learn more problems on : [Garbage Collections](http://www.indiabix.com/java-programming/garbage-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/garbage-collections/discussion-207) |

|  |  |
| --- | --- |
| 17. | public class X  {  public static void main(String [] args)  {  X x = new X();  X x2 = m1(x); /\* Line 6 \*/  X x4 = new X();  x2 = x4; /\* Line 8 \*/  doComplexStuff();  }  static X m1(X mx)  {  mx = new X();  return mx;  }  }  After line 8 runs. how many objects are eligible for garbage collection? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | 0 | | |  | B. | |  | | --- | | 1 | | |  | C. | |  | | --- | | 2 | | |  | D. | |  | | --- | | 3 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **B**  Explanation:  By the time line 8 has run, the only object without a reference is the one generated as a result of line 6. Remember that "Java is pass by value," so the reference variable x is not affected by the m1() method.  Ref: http://www.javaworld.com/javaworld/javaqa/2000-05/03-qa-0526-pass.html  Learn more problems on : [Garbage Collections](http://www.indiabix.com/java-programming/garbage-collections/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/garbage-collections/discussion-202) |

|  |  |
| --- | --- |
| 18. | public class Test  {  public void foo()  {  assert false; /\* Line 5 \*/  assert false; /\* Line 6 \*/  }  public void bar()  {  while(true)  {  assert false; /\* Line 12 \*/  }  assert false; /\* Line 14 \*/  }  }  What causes compilation to fail? |
| |  |  |  |  | | --- | --- | --- | --- | |  | A. | |  | | --- | | Line 5 | | |  | B. | |  | | --- | | Line 6 | | |  | C. | |  | | --- | | Line 12 | | |  | D. | |  | | --- | | Line 14 | |   Your Answer: Option **(Not Answered)**  Correct Answer: Option **D**  Explanation:  Option D is correct. Compilation fails because of an unreachable statement at line 14. It is a compile-time error if a statement cannot be executed because it is unreachable. The question is now, why is line 20 unreachable? If it is because of the assert then surely line 6 would also be unreachable. The answer must be something other than assert.  Examine the following:  A while statement can complete normally if and only if at least one of the following is true:  - The while statement is reachable and the condition expression is not a constant expression with value true.  -There is a reachable break statement that exits the while statement.  The while statement at line 11 is infinite and there is no break statement therefore line 14 is unreachable. You can test this with the following code:  public class Test80  {  public void foo()  {  assert false;  assert false;  }  public void bar()  {  while(true)  {  assert false;  break;  }  assert false;  }  }  Learn more problems on : [Assertions](http://www.indiabix.com/java-programming/assertions/)  Discuss about this problem : [Discuss in Forum](http://www.indiabix.com/java-programming/assertions/discussion-225) |