**Section I: Java Marks: 70**

|  |  |  |
| --- | --- | --- |
| Q1 | Which two of the following are legal declarations for non-nested 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 {} 7. 1 and 4 8. 2 and 5 9. 3 and 6 10. 4 and 6 | 2 |
| **Answer:** | C |  |
| Q2 | 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?     1. There is a syntax error on line 1. 2. There are syntax errors on lines 1 and 6. 3. There are syntax errors on lines 1, 6, and 8. 4. There is a syntax error on line 6. | 2 |
| Answer: | D |  |
| Q3 | 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?   1. Boo f = new Boo(24) { }; 2. Boo f = new Bar() { }; 3. Bar f = new Boo(String s) { }; 4. Boo f = new Boo.Bar(String s) { }; | 2 |
| Answer: | b |  |
| Q4 | public Object m() {  Object o = new Float(3.14F);  Object[] oa = new Object[l];  oa[0] = o; /\* Line 5 \*/  o = null; /\* Line 6 \*/  oa[0] = null; /\* Line 7 \*/  return o; /\* Line 8 \*/  }  When is the Float object, created in line 3, eligible for garbage  collection?   * 1. After line 5   2. After line 6   3. After line 7   4. After line 8 | 4 |
| Answer: | c |  |
| Q5 | 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?   * 1. 0   2. 1   3. 2   4. 3 | 4 |
| **Answer:** | b |  |
| Q6 | 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)  result = 1;  if (x.equals(y) )  result = result + 10;  if (x.equals(z) )  result = result + 100;  if (x.equals(x2) )  result = result + 1000;  if (x.equals(z2) )  result = result + 10000;  System.out.println("result = " + result);  }  }  What is the output of above program | 4 |
| **Answer:** | 10 |  |
| Q7 | 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 \*/  What is the output of the above program?   * 1. An exception occurs at runtime at line 10.   2. It prints "Zippo".   3. Compilation fails because of an error on line 7.   4. Compilation fails because of an error on line 13. | 4 |
| **Answer:** | b |  |
| Q8 | 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());  }  } | 4 |
| **Answer:** | 57 22 |  |
| Q9 | 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 + "..");  }  }  } | 4 |
| **Answer:** | 1..2.. |  |
| Q10 | public class If2 {  static boolean b1, b2;  public static void main(String[] args) {  int x = 0;  if (!b1) /\* Line 7 \*/  {  if (!b2) /\* Line 9 \*/  {  b1 = true;  x++;  if (5 > 6) {  x++;  }  if (!b1)  x = x + 10;  else if (b2 = true) /\* Line 19 \*/  x = x + 100;  else if (b1 | b2) /\* Line 21 \*/  x = x + 1000;  }  }  System.out.println(x);  }  }  What will be the output of above program?    a. 0  b. 101  c. 1  d. 111 | 4 |
| **Answer:** | b |  |
| Q11 | 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 \*/  }  }  What will be the output of above program? | 4 |
| **Answer:** | BaseBase |  |
| Q12 | 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. 7. 1 and 3 8. 2 and 4 9. 1 and 5 10. 2 and 6 | 2 |
| **Answer:** | d |  |
| Q13 | 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. | 4 |
| **Answer:** | c |  |
| Q14 | 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();  }  } | 4 |
| **Answer:** | BCD |  |
| Q15 | Which interface does java.util.Hashtable implement?   1. java.util.Map 2. java.util.List 3. java.util.HashTable 4. java.util.Collection | 2 |
| **Answer:** | a |  |
| Q16 | package foo;  import java.util.Vector; /\* Line 2 \*/  private class MyVector extends Vector /\* Line 3\*/  {  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 \*/  }  }  What will be output of above program?   1. Compilation will succeed. 2. Compilation will fail at line 3. 3. Compilation will fail at line 5. 4. Compilation will fail at line 15. | 2 |
| **Answer:** | b |  |
| Q17 | 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 \*/  }  }  Which line causes compilation to fail?   1. Line 5 2. Line 6 3. Line 12 4. Line 14 | 2 |
| **Answer:** | d |  |
| Q18 | 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(); 9. 1,2,4 10. 2,4,5 11. 1,2,6 12. 2,3,4 | 2 |
| **Answer:** | c |  |
| Q19 | class Voop {  public static void main(String [] args) {  doStuff(1);  doStuff(1,2);  }  // insert code here  }  Which, inserted independently at line 6, will compile? (Choose all that apply.)   * 1. static void doStuff(int... doArgs) { }   2. static void doStuff(int[] doArgs) { }   3. static void doStuff(int doArgs...) { }   4. static void doStuff(int... doArgs, int y) { }   5. static void doStuff(int x, int... doArgs) { } | 2 |
| **Answer:** | a, e |  |
| Q20 | What is the output for the below code?  String a = "newspaper";  a = a.substring(5,7);  char b = a.charAt(1);  a = a + b;  System.out.println(a); | 2 |
| **Answer:** | app |  |
| Q21 | 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);  }  } | 2 |
| **Answer:** | javajavac |  |
| Q22 | Which three form part of correct array declarations?   1. public int a [ ] 2. static int [ ] a 3. public [ ] int a 4. private int a [3] 5. private int [3] a [ ] 6. public final int [ ] a 7. 1,3,4 8. 2,4,5 9. 1,2,6 10. 2,5,6 | 2 |
| **Answer:** | c |  |
| Q23 | public class OverridingTest {  public static void main(String...args){  AA obj = new AAAA();  int value = obj.getResult();  System.out.println(value);  }  }  class AA{  final public int getResult(){  return 1;  }  }  class AAAA extends AA{  public int getResult(){  return 2;  }  }  What will be output of above program?   * 1. Compilation Fails   2. 2   3. 1 | 2 |
| **Answer:** | a |  |
| Q24 | 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. 7. 1 and 2 8. 3 and 5 9. 4 and 6 10. 1 and 3 | 2 |
| **Answer:** | c |  |
| Q25 | Which statement is true?   1. A static method cannot be synchronized. 2. If a class has synchronized code, multiple threads can still access the nonsynchronized code. 3. Variables can be protected from concurrent access problems by marking them with the synchronized keyword. 4. When a thread sleeps, it releases its locks. | 2 |
| **Answer:** | b |  |