|  |
| --- |
| **PART- A**  **Advance Programming RollNo:**  End Semester Exam-2017-18 |

**Time: 1.0 hrs Date:** 29/11/2017 **Max. Marks:** 20

**Q1A. Correct the compile time error (if any) and find the output of the following Java codes:**

**(Also show the workout in the given space otherwise the answer will not be evaluated)**

|  |  |
| --- | --- |
| **QA**.1 **[1]**  class Output  { public static void main(String args[])  {  Integer i = new Integer(257);  byte x = i.byteValue();  System.out.print(x);  }  } | Byte variable is one byte so can hold only -128 to +127  So if we convert 257 into byte value then it is 1  Output: **1** |
| **QA.2** **[2]**  class TreeDemo{  public static void main(String args[])  { TreeSet<String> tmap = new TreeSet<String>();  tmap.add("one");  tmap.add("two");  tmap.add("three");  tmap.add("four");  tmap.add("five");  tmap.add(“one”);  Iterator it = tmap.iterator();  while (it.hasNext() )  {  System.out.print( it.next() + " " );  }  }} | TreeSet followed the natural order of the values so in this case alphabetical order tree will be generated.  The resultant tree is balanced binary search tree:  Last statement  tmap.add(“one”);  will not be added because treeset doesnot contain duplicate hence  **Output will be: five four one three two** |
| **QA.3** **[1]** class Output {  public static void main(String args[])  {  Double d2 = new Double("-5.5");  Double d3 = new Double("-5.5");  System.out.println(d2==d3);  System.out.println(d2.equals(d3));  **}**  **}** | **Both the objects have different memory locations**  **But have same value. So:**  System.out.println(d2==d3);  **Check the references so output is:false**  System.out.println(d2.equals(d3));  **Checks the values so output is: true** |
| **QA.4 [1]**  class Output  { public static void main(String args[])  {Boolean b1 = new Boolean("TRUE");  Boolean b2 = new Boolean("true");  Boolean b3 = new Boolean("JUNK");  System.out.println(b1 + b2 + b3);  }**}** | **First there is a syntax error in**  System.out.println(b1 + b2 + b3);  To remove syntax error we have to write so that the expression will be string  System.out.println**(“ “** +b1 + b2 + b3);  Then output is: **truetruefalse**  The third JUNK will not be printed |
| **QA.5 [2]**  import java.util.\*;  public class Names  { static String s;  public Names(String s)  {  Names.s = s;  }  public static void main(String[] args)  {  HashSet<Object> hs = new HashSet<Object>();  hs.add(new Names("pranavi"));  hs.add(new Names("madhavi"));  hs.add(new Names("jahnavi"));  hs.add(new String("vyshnavi"));  hs.add(new String("vyshnavi"));  System.out.println(hs.size());  System.out.println(hs);  }  } | **In these set of statements:**  hs.add(new Names("pranavi"));  hs.add(new Names("madhavi"));  hs.add(new Names("jahnavi"));  References of type Names are stored not the strings  hs.add(new String("vyshnavi"));  hs.add(new String("vyshnavi"));  in these two statements Strings are same so Hashset cant have duplicates so only one name is added to HashSet:  System.out.println(hs.size());  System.out.println(hs);    Output: **4**  [refence1, refence2, refence3, vyshnavi]  **Order may be any because HashSet does not maintain order.** |
| **QA.6 [2]**  import java.util.\*;  class Arraylist {  public static void main(String args[])  { ArrayList<String> obj = new ArrayList<String>();  obj.add("A");  obj.add("B");  obj.add("C");  obj.add(1, "D");  obj.add("E");  obj.add(1,"F");  obj.remove(3);;  obj.add(4,"B");  System.out.println(obj +" " + obj.size());  }  } | obj.add("A");  obj.add("B");  obj.add("C");  After three statements Arraylist will be**:[A, B, C]**  Because of this statement: **obj.add(1, "D");**  D will be added at position 1:  So after that Arraylist will be: **[A, D, B, C]**  Because of this statement: **obj.add("E");**  Arraylist will be**:[A, D, B, C, E]**  Because of this statement: **obj.add(1, "F");**  Arraylist will be**:[A, F,D, B, C, E]**  Because of this statement: **obj.remove(3);**  Arraylist will be**:[A, F,D, C, E]**  Because of this statement: **obj.add(4, "B");**  Arraylist will be**:[A, F, D, C, B, E]**  Because of this statement:  **System.out.println(obj +" " + obj.size());**  Output will be:**[A, F, D, C, B, E] 6 (**size of Array) |