ViperWolf @ Oracle

Invitational

February 10, 2018

General Directions:

1) DO NOT OPEN EXAM UNTIL TOLD TO DO SO.

2) **NO CALCULATORS of any kind may be used.**

3) You have 45 minutes to complete this contest. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.

4) Papers may not be turned in until forty-five minutes have elapsed. If you finish the test before the end of the allotted time, remain at your seat and retain your paper until told to do otherwise. You may use this time to check your answers.

5) All answers must be written on the answer sheet/Scantron card provided. Indicate your answers in the appropriate blanks provided on the answer sheet or on the Scantron card. Clean erasures are necessary for accurate Scantron grading.

6) You may place as many notations as you desire anywhere on the test paper except on the answer sheet or Scantron card which is reserved for answers only.

7) You may use additional scratch paper provided by the contest director.

8) All questions have ONE and only ONE correct (BEST) answer. There is a penalty for all incorrect answers. **All provided code segments are intended to be syntactically correct, unless otherwise stated (i.e. error is an answer choice). Ignore any typographical errors and assume any undefined variables are defined as used.**

9) A reference to commonly used Java classes is provided with the test and you may use this reference during the contest. You may detach the reference sheets from the test booklet but DO NOT DO SO UNTIL THE CONTEST BEGINS.

10) Assume that any necessary import statements for Standard Java 2 Packages and classes (e.g. .lang, .util, System, Math, Double, etc.) are included in any programs or code segments that refer to methods from these classes and/or packages.

Scoring:

1. All questions will receive 6 points if answered correctly; no points will be given or subtracted if unanswered; 2 points will be deducted for each incorrect answer.

This Page Intentionally Left Blank – Use as Scratch Paper

**Standard Classes and Interfaces — Supplemental Reference**

|  |  |
| --- | --- |
| **class java.lang.Object**   * boolean equals(Object other) * String toString() * int hashCode()   **interface java.lang.Comparable<T>**   * int compareTo(T other) Return value < 0 if this is less than other. Return value = 0 if this is equal to other. Return value > 0 if this is greater than other.   **class java.lang.Integer implements**  **Comparable<Integer>**   * Integer(int value) * int intValue() * boolean equals(Object obj) * String toString() * int compareTo(Integer anotherInteger) * static int parseInt(String s)   **class java.lang.Double implements**  **Comparable<Double>**   * Double(double value) * double doubleValue() * boolean equals(Object obj) * String toString() * int compareTo(Double anotherDouble) * static double parseDouble(String s)   **class java.lang.String implements**  **Comparable<String>**   * int compareTo(String anotherString) * boolean equals(Object obj) * int length() * String substring(int begin, int end) Returns the substring starting at index begin and ending at index (end - 1). * String substring(int begin) Returns substring(from, length()). * int indexOf(String str) Returns the index within this string of the first occurrence of str. Returns -1 if str is not found. * int indexOf(String str, int fromIndex) Returns the index within this string of the first occurrence of str, starting the search at the specified index.. Returns -1 if str is not found. * charAt(int index) * int indexOf(int ch) * int indexOf(int ch, int fromIndex) * String toLowerCase() * String toUpperCase() * String[] split(String regex) * boolean matches(String regex)     **class java.util.Stack<E>**   * boolean isEmpty() * E peek() * E pop() * E push(E item)   **interface java.util.Queue<E>**   * boolean add(E e) * boolean isEmpty() * E peek() * E remove()   **class java.util.PriorityQueue<E>**   * boolean add(E e) * boolean isEmpty() * E peek() * E remove()   **interface java.util.Set<E>**   * boolean add(E e) * boolean contains(Object obj) * boolean remove(Object obj) * int size() * Iterator<E> iterator() * boolean addAll(Collection<? extends E> c) * boolean removeAll(Collection<?> c) * boolean retainAll(Collection<?> c)   **class java.util.HashSet<E> implements Set<E>  class java.util.TreeSet<E> implements Set<E>  interface java.util.Map<K,V>**   * Object put(K key, V value) * V get(Object key) * boolean containsKey(Object key) * int size() * Set<K> keySet() * Set<Map.Entry<K, V>> entrySet()   **class java.util.HashMap<K,V> implements Map<K,V>  class java.util.TreeMap<K,V> implements Map<K,V>**  **interface java.util.Map.Entry<K,V>**   * K getKey() * V getValue() * V setValue(V value)   **interface java.util.Iterator<E>**   * boolean hasNext() * E next() * void remove()   **interface java.util.ListIterator<E> extends**  **java.util.Iterator<E>**  Methods in addition to the Iterator methods:   * void add(E e) * void set(E e) | **class java.lang.Character**   * static boolean isDigit(char ch) * static boolean isLetter(char ch) * static boolean isLetterOrDigit(char ch) * static boolean isLowerCase(char ch) * static boolean isUpperCase(char ch) * static char toUpperCase(char ch) * static char toLowerCase(char ch)   **class java.lang.Math**   * static int abs(int a) * static double abs(double a) * static double pow(double base,   double exponent) * static double sqrt(double a) * static double ceil(double a) * static double floor(double a) * static double min(double a, double b) * static double max(double a, double b) * static int min(int a, in b) * static int max(int a, int b) * static long round(double a) * static double random() Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.   **interface java.util.List<E>**   * boolean add(E e) * int size() * Iterator<E> iterator() * ListIterator<E> listIterator() * E get(int index) * E set(int index, E e) Replaces the element at index with the object e. * void add(int index, E e) Inserts the object e at position index, sliding elements at position index and higher to the right (adds 1 to their indices) and adjusts size. * E remove(int index) Removes element from position index, sliding elements at position (index + 1) and higher to the left (subtracts 1 from their indices) and adjusts size.   **class java.util.ArrayList<E> implements List<E>**  **class java.util.LinkedList<E> implements**  **List<E>, Queue<E>**  Methods in addition to the List methods:   * void addFirst(E e) * void addLast(E e) * E getFirst() * E getLast() * E removeFirst() * E removeLast()   **class java.lang.Exception**   * Exception() * Exception(String message)   **class java.util.Scanner**   * Scanner(InputStream source) * boolean hasNext() * boolean hasNextInt() * boolean hasNextDouble() * String next() * int nextInt() * double nextDouble() * String nextLine() * Scanner useDelimiter(String pattern) |

Computer Science Contest #1718-12 February 10, 2018

Note: Correct responses are based on Java, **J2sdk v 1.8.x**, from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (i. e. error is an answer choice) and any necessary Java 2 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used. **For all output statements, assume that the System class has been statically imported… *import static java.lang.System.\*;***

|  |  |
| --- | --- |
| Question 1 xx  What is 3016 plus34916 ?  A. 92110 B. 11101110012 C. 38B16 D. 88910 E. 11001110112 | |
| Question 2 xx  What is output by the code to the right?  A. -159.5 B. 0.0 C. -215.0  D. -214.837 E. -160.0 | int x = 43, y = x - 8, k = 27;  double z = x / 2 + y - k \* (x - y);  out.println(z); |
| Question 3 xx  Which of the following correctly replaces <\*1> in the code to the right such that the output is  212 \ CAT\ 4.62  01234567890123456789  A. out.printf("%4d\%5s\%7.2f\n%s",a,c,b,x)  B. out.printf("%-4d\\%5s\\%7.2f\n%s",a,c,b,x)  C. out.printf("%4d\\%5s\\%-7.2d\n%s",a,c,b,x)  D. out.printf("%4d\%5s\%-7.2f\n%s",a,b,c,x)  E. out.printf("%-4d\%5s\%7.2f%s",a,c,b,x) | String x = "0123456789012345";  int a = 212;  double b = 4.618;  String c = "CAT";  <\*1>; |
| Question 4 xx  What can replace <\*1> and <\*2> in the code to the right such that the String y is the substring between the first and last appearance of the character T?  A. int k = x.indexOf("T")+1;  int p = x.lastIndexOf("T")+1;  B. int k = x.indexOf("T");  int p = x.lastIndexOf("T")+1;  C. int k = x.indexOf("T")+1;  int p = x.lastIndexOf("T")-1;  D. int k = x.indexOf("T");  int p = x.lastIndexOf("T")-1;  E. int k = x.indexOf("T")+1;  int p = x.lastIndexOf("T"); | String x = "TAGCCGCAACGTCAGGT";  int k = <\*1>;  int p = <\*2>;  String y = x.substring(k,p); |
| Question 5 xx  Which of the following is an example of DeMorgan’s Law?  I. !A || !B = !(A && B)  II. !A || B = !(A && !B)  III. A && !B = !(!A || B)  A. I only B. II and III C. I and III D. I and II E. I, II, and III | |
| Question 6 xx  What is the correct code that will randomly generate a number between -5 inclusively and 46 inclusively?  A. int x = (int)(Math.random(51) - 5);  B. int x = (int)(Math.random()\*52 - 5);  C. int x = (int)(Math.random(-5, 47));  D. int x = (int)(Math.random(-5, 46));  E. int x = (int)(Math.random()\*51 - 5); | |
| Question 7 xx  What is output by the code to the right?  A. 10.0  B. 9.9  C. 132.0  D. 242.0  E. 9.8 | double x = 11.0;  double sum = 0;  while(x>10)  {  sum += x\*2+1;  x-=.1;  }  out.println(x); |
| Question 8 xx  How many Zs are printed by the code to the right?  A. 4  B. 3  C. 0  D. 2  E. 1 | String str = "ABCBA";  String y = "";  for(int i=0; i<str.length(); i++)  {  switch(str.charAt(i))  {  case 'A': y += "B";  case 'B': y += "Z";  case 'C': y += "A";  default: y+= "X";  }  }  out.println(y); |
| Question 9 xx  What letter is printed out the most by the code to the right regardless of what the object of str is?  A. B  B. X  C. Z  D. A  E. It is dependent on the value of str. |
| Question 10 xx  What is output by the code to the right?  A. 98  B. 87  C. 88  D. 79  E. There is no output due to a run-time error | int[] list = {2,9,2,2,3,1,1,2};  int sum = 100;  for(int i=0; i<list.length; i++)  sum-=list[list[i]];  out.println(sum); |
| Question 11 xx  What is output by the code to the right assuming that Hansberry.txt links to the above Hansberry.txt file?  A. that that  B. that say  C. well that  D. that me?  E. There is no output due to a run-time error | /////Hansberry.txt////  //you you sent all the  //say home for me?  //for you i would do  //much more well that  Scanner input =  new Scanner(new File("Hansberry.txt"));  Scanner reput =  new Scanner(new File("Hansberry.txt"));  int k = 0;  String x="",y="";  while(input.hasNext())  {  if(k%3==0)  y=reput.next();  x=input.next();  k++;  }  out.println(x+" "+y); |
| Question 12 xx  What is output by the code to the right?  A. 175  B. 136  C. 96  D. 80  E. 108 | int[] x = {12,38,45,13,28,30,9};  int sum = 0;  for(int i=x.length/2; i>=0; i--)  sum += x[i];  out.println(sum); |
| Question 13 xx  What is the order of precedence from lowest to highest from the choices to the right?  A. I, II, III, IV, V  B. IV, II, I, III, V  C. IV, II, V, III, I  D. II, V, IV, III, I  E. II, I, III, V, IV | I. ++  II. ^  III. +  IV. ==  V. << |
| Question 14 xx  Which of the following data types can store the long primitive data type?  A. int B. char C. boolean D. double E. none of these | |
| Question 15 xx  What is output by the code to the right?  A. rock bye baby you this about  B. songs you sweet rock  C. rock bye baby you songs think  D. rock sweet you this  E. rock sweet you songs | String x,y;  x = "rock a bye sweet baby james";  y = "you think this songs about you";  ArrayList<String> list =  new ArrayList<String>();  Scanner input = new Scanner(x);  while(input.hasNext())  list.add(input.next());  input = new Scanner(y);  int count = list.size();  while(input.hasNext())  list.add(count,input.next());  String z = "";  for(int i=0; i<list.size(); i+=2)  z+=list.remove(i)+" ";  out.println(z); |
| Question 16 xx  What is output at //<\*1> by the code to the right?  A. [SAMMY, PETER, BILLY]  B. [PETER, BILLY, SAMMY]  C. [SAMMY, BILLY, PETER]  D. [BILLY, SAMMY, PETER]  E. [BILLY, PETER, SAMMY] | class K implements Comparable<K> {  private String str;  public K(String x)  { str = x; }  public String mys()  { int x = str.length()/2;  return str.substring(x) +  str.substring(0,x);  }  public int compareTo(K x)  {  if(str.length()!=x.str.length())  return str.length() -  x.str.length();  if(str.length()==1)  return str.compareTo(x.str);  return mys().compareTo(x.mys());  }  public String toString()  { return str; }  }  //CLIENT CODE  ArrayList<K> x = new ArrayList<>();  x.add(new K("SAMMY"));  x.add(new K("BILLY"));  x.add(new K("PETER"));  Collections.sort(x);  out.println(x); //<\*1>  x.clear();  x.add(new K("I"));  x.add(new K("WANT"));  x.add(new K("A"));  x.add(new K("NEW"));  x.add(new K("DUCK"));  x.add(new K("ONE"));  x.add(new K("THAT"));  x.add(new K("WONT"));  Collections.sort(x);  out.println(x); //<\*2> |
| Question 17 xx  What is output by the code to the right?  A. [A, I, NEW, ONE, DUCK, THAT, WANT, WONT]  B. [A, I, NEW, ONE, WANT, THAT, WONT, DUCK]  C. [A, I, NEW, ONE, THAT, DUCK, WANT, WONT]  D. [A, DUCK, I, NEW, ONE, THAT, WANT, WONT]  E. [I, WANT, A, NEW, DUCK, ONE, THAT, WONT] |
| Question 18 xx  What is output by the code on the right?  A. 206  B. 244  C. 236  D. 241  E. There is no output due to a run-time error | int[] x = {38,17,45,13,32,28,46,49};  ArrayList<Integer> list =  new ArrayList<>();  for(int k:x)  list.add(k);  int i = 1;  int temp = list.get(0);  while(i<list.size())  {  if(i%3==0)  list.remove(i);  temp+=list.get(i);  i++;  }  out.println(temp); |
| Question 19 xx  What is assigned to the value of x and y such that the code on the right outputs:  2 2 16  A. x = 232; y = 16;  B. x = 216; y = 32;  C. x = 200; y = 35;  D. x = 250; y = 6;  E. The answer cannot be determined with the given information | int a,b,c;  int x = ???;  int y = ???;  c = 512;  a = c / x;  c %= x;  b = c / y;  c %= y;  out.print(a+" "+b+" "+c); |
| Question 20 xx  What is output by the code to the right?  A. MCGUFFINCGUFFIGUFFUF  B. MCGUFFIN  C. CGUFFIGUFFUF  D. MCGUFFINCGUFFIGUFF  E. There is no output due to a run-time error | String x = "MCGUFFIN";  int loc=0;  do  {  out.print(x);  loc = x.length();  x = x.substring(1,loc-1);  }while(x.length()/2>1); |
| Question 21 xx  What is output on line <\*1> by the code on the right?  A. 24.0  B. 26.25  C. -1  D. 25.0  E. 15.61 | public static double myst1(double x,  double y)  {  double d = x-y;  double k = d/y;  if(k>.5)  return x\*.1 + d\*.01;  else if(k>=.25)  return x\*.15 + d\*.03;  else if(k>=.1)  return x\*.2 + d\*.05;  else if(k>=0)  return x\*.25;  else  return -1;  }  // CLIENT CODE  out.println(myst1(150,100)); //<\*1>  out.println(myst1(430,300)); //<\*2> |
| Question 22 xx  What is output on line <\*2> by the code on the right?  A. -1  B. 68.4  C. 45.8  D. 107.5  E. 45.3 |
| USE THE FOLLOWING FOR QUESTIONS 23-26  class B {  private int[] stats;  public B(int []temp)  {  stats = new int[temp.length];  for(int i=0; i<temp.length; i++)  stats[i]=temp[i];  }    public int get(int x)  { return stats[x]; }    public void change(int loc, int v)  {  stats[loc]+=v;  }    public String toString()  {  String x = "";  for(int k:stats)  x+=k+" ";  return x;  }  }  class P extends B  {  private int[] other;  public P(int[] x, int[] y)  {  super(x);  other = new int[y.length];  for(int i=0; i<y.length; i++)  other[i]=y[i];  }    public int get(boolean isK, int x)  {  if(isK)  return other[x];  else  return super.get(x);  }    public void change(int loc, int v)  {  other[loc]+=v;  super.change(loc, -v);  }    public String toString()  {  String x = super.toString();  for(int k:other)  x+=k+" ";  return x;  }  } | // Client Code  int[] a = {1,7,9,5};  int[] b = {0,4,3,9};  int[] c = {3,1,4,8};  int[] d = {4,6,8,7};  B one = new B(a);  B two = new P(b,c);  P three = new P(d,a);  B four = new B(b);  out.println(four); //<\*1>  out.println(two); //<\*2>  B[] list = new B[4];  list[0] = one;  list[1] = two;  list[2] = three;  list[3] = four;  int num = 0;  int q, r;  for(int i=0; i<list.length-1; i++)  {  for(int j=i+1; j<list.length; j++)  {  if(list[i] instanceof P)  {  P temp = (P)(list[i]);  q = temp.get(num%4);  }  else  q = list[i].get(num%4);  r = list[j].get(num%4);  q = q-r;  list[i].change(num%4,q);  list[j].change(num%4,q);  if(num == 1)  out.println(list[0]); //<\*3>  num++;  }  }  out.println(list[3]); //<\*4> |
| Question 23 xx  What is output on line //<\*1> by the code on the previous page?  A. 1 7 9 5 3 1 4 8  B. 0 4 3 9  C. 1 7 9 5  D. 3 1 4 8  E. 4 6 8 7 | |
| Question 24 xx  What is output on line //<\*2> by the code on the previous page?  A. 3 1 4 8 0 4 3 9  B. 0 4 3 9  C. 1 7 9 5  D. 0 4 3 9 3 1 4 8  E. 3 1 4 8 | |
| Question 25 xx  What is output on line //<\*3> by the code on the previous page?  A. 2 8 9 5  B. -1 4 3 9 4 1 4 8  C. -1 4 9 9  D. 4 5 8 5 1 8 9 7  E. 2 8 15 5 | |
| Question 26 xx  What is output on line //<\*4> by the code on the previous page?  A. -1 5 9 9  B. 0 4 3 7 3 1 4 10  C. 0 4 3 9  D. 2 8 15 5  E. 4 4 8 5 1 9 9 7 | |
| Question 27 xx  What is returned by the call mys2(22)?  A. -16  B. -12  C. -14  D. -54  E. -24 | public static int myst2(int k)  {  if (k<=5 && k>=-10)  return k\*2;  if (k>5)  return 2+myst2(k-3);  return -2+myst2(k+4);  } |
| Question 28 xx  What is returned by the call mys2(22)?  A. 20  B. 16  C. 24  D. 22  E. 8 |
| Question 29 xx  What is 7ABC16?  A. 11110101011112  B. 1111010101111002  C. 710111210  D. 3142110  E. 1100101110101112 | |
| Question 30 xx  What would a binary search tree look like if the following values was entered into the tree in the following order?  21 44 16 40 29 17 13  A. 13 B. 21  16 17 16 44  21 29 40 44 13 17 40  29  C. 21 D. 21  16 44 16 29  13 29 40 13 17 40 44  17  E. 21  44 16  40 29 17 13 | |
| Question 31 xx  What is output on line //<\*1> by the code on the right?  A. {I=[M, P, S], P=[I, P], S=[I, S]}  B. {I, M, P, S}  C. {[S, S, P], [I], [P, I], [S, I, S, I]}  D. {I=[S, S, P], M=[I], P=[P, I],  S=[S, I, S, I]}  E. There is no output due to a run time error. | String str = "MISSISSIPPI";  TreeMap<String,ArrayList<String>> m =  new TreeMap<>();  for(int i=0; i<str.length()-1; i++)  {  String x = str.substring(i,i+1);  String y = str.substring(i+1,i+2);  if(!m.containsKey(x))  m.put(x,new ArrayList<String>());  m.get(x).add(y);  }  *out*.println(m); //<\*1>  ArrayList<String> temp = m.remove("I");  Iterator<String> iter =  m.keySet().iterator();  while(iter.hasNext())  if(temp.isEmpty())  m.get(iter.next()).add("X");  else  m.get(iter.next()).add(temp.remove(0));  *out*.println(m); //<\*2> |
| Question 32 xx  What is output on line //<\*2> by the code on the right?  A. {[I, S], [P, I, S], [S, I, S, I, p]}  B. {M=[I, S], P=[P, I, S], S=[S, I, S, I, P]}  C. {M=[I, x], P=[P, I, S], S=[S, I, S, I, x]}  D. {[I, X], [P, I, S], [S, I, S, I, X]}  E. There is no output due to a run time error. |
| Question 33 xx  What is output by the code to the right?  A. [47, 33, 40]  B. [40, 33, 47]  C. [40, 47, 33]  D. [33]  E. There is no output due to run time error | Queue<Integer> q = new LinkedList<Integer>();  q.offer(21);  q.offer(44);  q.offer(16);  q.poll();  q.offer(40);  q.peek();  q.offer(47);  q.poll();  q.poll();  q.offer(q.poll());  q.offer(33);  q.offer(q.peek());  q.poll();  out.println(q); |
| Question 34 xx  What is output by the code to the right?  A. -63  B. 16777212  C. -64  D. -16777212  E. There is no output due to run time error | out.println(Integer.MAX\_VALUE<<6); |
| Question 35 xx  Which value of str would case the code to the right to print false?  A. blood  B. belad  C. bead  D. boald  E. baleed | boolean x;  x=str.matches("b.\*(e|l|o).\*(e|l|o).\*d");  out.println(x); |
| Question 36 xx  Simplify the Boolean algebra statement on the right  A. A!C  B. A + B + !C  C. B  D. A + !B  E. true | !(!B(!(AB)+!AC)) |
| Question 37 xx  Solve the following expression?  RSHIFT-4(10010111 OR (LCIRC-3 00100111 AND LSHIFT-2 01010011))  A. 10010111  B. 11110000  C. 10010111  D. 00001001  E. 00001000 | |
| Question 38 xx  Convert the prefix notation equation to the right into an infix notation equation.  A. A/B-C/D\*E+F  B. A/(B-C)/(D\*(E+F))  C. (A/B-C)/D\*(E+F)  D. A/B/(C\*D-(E+F))  E. (A-B/C\*D+E)/F | //A-BC\*D+EF |
| Question 39 xx  *OPEN ENDED QUESTION – Find the answer and write it on your answer sheet. If you are using a ScanTron form, write the question number and the answer on the bottom of the ScanTron.*  What is the decimal value of the number to the right if it is being stored as a signed byte data type? | 101010102 |
| *Question 40 xx*  *OPEN ENDED QUESTION – Find the answer and write it on your answer sheet. If you are using a ScanTron form, write the question number and the answer on the bottom of the ScanTron.*  *Write all the ordered triplets that will make the digital circuit diagram to the right false. Write a triplet in the form of (A, B, C), where A, B, and C will either be a 0 or a 1. For example: (1,1,1) or (0,1,1).* | A  B  A  C |