UIL Computer Science Written Test

2019 Invitational A

January/February 2019

General Directions (Please read carefully!)

1. DO NOT OPEN THE EXAM UNTIL TOLD TO DO SO.
2. There are 40 questions on this contest exam. You will have 45 minutes to complete this contest.
3. All answers must be legibly written on the answer sheet provided. Indicate your answers in the appropriate blanks provided on the answer sheet. Clean erasures are necessary for accurate grading.
4. You may write on the test packet or any additional scratch paper provided by the contest director, but NOT on the answer sheet, which is reserved for answers only.
5. All questions have ONE and only ONE correct answer. There is a 2-point penalty for all incorrect answers.
6. Tests may not be turned in until 45 minutes have elapsed. If you finish the test before the end of the allotted time, remain at your seat and retain your test until told to do otherwise. You may use this time to check your answers.
7. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.
8. All provided code segments are intended to be syntactically correct, unless otherwise stated. You may also assume that any undefined variables are defined as used.
9. A reference to many commonly used Java classes is provided with the test, and you may use this reference sheet during the contest. AFTER THE CONTEST BEGINS, you may detach the reference sheet from the test booklet if you wish.
10. Assume that any necessary import statements for standard Java SE packages and classes (e.g., `java.util`, `System`, etc.) are included in any programs or code segments that refer to methods from these classes and packages.
11. NO CALCULATORS of any kind may be used during this contest.

Scoring

1. Correct answers will receive **6 points**.
2. Incorrect answers will lose **2 points**.
3. Unanswered questions will neither receive nor lose any points.
4. In the event of a tie, the student with the highest percentage of attempted questions correct shall win the tie.

STANDARD CLASSES AND INTERFACES – SUPPLEMENTAL REFERENCE

package java.lang

```
class Object
  boolean equals(Object anotherObject)
  String toString()
  int hashCode()

interface Comparable<T>
  int compareTo(T anotherObject)
    Returns a value < 0 if this is less than anotherObject.
    Returns a value = 0 if this is equal to anotherObject.
    Returns a value > 0 if this is greater than anotherObject.

class Integer implements Comparable<Integer>
  Integer(int value)
  int intValue()
  boolean equals(Object anotherObject)
  String toString()
  String toString(int i, int radix)
  int compareTo(Integer anotherInteger)
  static int parseInt(String s)

class Double implements Comparable<Double>
  Double(double value)
  double doubleValue()
  boolean equals(Object anotherObject)
  String toString()
  int compareTo(Double anotherDouble)
  static double parseDouble(String s)

class String implements Comparable<String>
  int compareTo(String anotherString)
  boolean equals(Object anotherObject)
  int length()
  String substring(int begin)
    Returns substring(begin, length()).
  String substring(int begin, int end)
    Returns the substring from index begin through index (end - 1).
  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 fromIndex. Returns -1 if str is not found.
  int indexOf(int ch)
  int indexOf(int ch, int fromIndex)
  char charAt(int index)
  String toLowerCase()
  String toUpperCase()
  String[] split(String regex)
  boolean matches(String regex)
  String replaceAll(String regex, String str)

class 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 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, int b)
  static int max(int a, int b)
  static long round(double a)
  static double random()
    Returns a double greater than or equal to 0.0 and less than 1.0.
```

package java.util

```
interface List<E>
class ArrayList<E> implements List<E>
  boolean add(E item)
  int size()
  Iterator<E> iterator()
  ListIterator<E> listIterator()
  E get(int index)
  E set(int index, E item)
  void add(int index, E item)
  E remove(int index)

class LinkedList<E> implements List<E>, Queue<E>
  void addFirst(E item)
  void addLast(E item)
  E getFirst()
  E getLast()
  E removeFirst()
  E removeLast()

class Stack<E>
  boolean isEmpty()
  E peek()
  E pop()
  E push(E item)

interface Queue<E>
class PriorityQueue<E>
  boolean add(E item)
  boolean isEmpty()
  E peek()
  E remove()

interface Set<E>
class HashSet<E> implements Set<E>
class TreeSet<E> implements Set<E>
  boolean add(E item)
  boolean contains(Object item)
  boolean remove(Object item)
  int size()
  Iterator<E> iterator()
  boolean addAll(Collection<? extends E> c)
  boolean removeAll(Collection<?> c)
  boolean retainAll(Collection<?> c)

interface Map<K,V>
class HashMap<K,V> implements Map<K,V>
class TreeMap<K,V> implements 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()

interface Iterator<E>
  boolean hasNext()
  E next()
  void remove()

interface ListIterator<E> extends Iterator<E>
  void add(E item)
  void set(E item)

class Scanner
  Scanner(InputStream source)
  Scanner(String str)
  boolean hasNext()
  boolean hasNextInt()
  boolean hasNextDouble()
  String next()
  int nextInt()
  double nextDouble()
  String nextLine()
  Scanner useDelimiter(String regex)
```

UIL COMPUTER SCIENCE WRITTEN TEST – 2019 INVITATIONAL A

Note: Correct responses are based on **Java SE Development Kit 8 (JDK 8)** from Oracle, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g., "error" is an answer choice) and any necessary Java SE 8 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 using: `import static java.lang.System.*;`**

Question 1.

What is the decimal equivalent of $2AB_{16}$?

A) 700 **B)** 667 **C)** 342 **D)** 751 **E)** 683

Question 2.

What is the output of the code segment to the right?

A) 34 **B)** 18 **C)** -24 **D)** 50 **E)** 46

```
out.print(45-8*2/3+-6);
```

Question 3.

What is the output of the code segment to the right?

A)
```
Socks
Bubba
Binky
```

B) SocksBubbaBinky

C)
```
Socks
BubbaBinky
```

D)
```
SocksBubba
Binky
```

E) Socks Bubba Binky

```
out.println("Socks");
out.print("Bubba");
out.println("Binky");
```

Question 4.

What is the output of the code segment to the right?

A) defghab **B)** efghbc **C)** defghabc

D) efghbcd **E)** fghbcd

```
String str="abcdefgh";
out.print(str.substring(4)+
          str.substring(1, 3));
```

Question 5.

Which of the lines shown on the right will print `false`?

A) line #1

B) line #2

C) line #3

D) line #4

E) More than one of the above.

```
boolean a=true,b=true,c=true;
out.println(a&&b&&c);//line #1
out.println(a&&b||c);//line #2
out.println(a&&b^c);//line #3
out.println(a||b^c);//line #4
```

Question 6.

What is the output of the code segment to the right?

A) 3 **B)** 3.0 **C)** 4 **D)** 4.0 **E)** 3.2

```
double x=Math.PI;
double y=Math.ceil(x);
out.print(y);
```

Question 7.

What is the output of the code segment to the right?

A) 10.875 **B)** 3.375 **C)** 10 **D)** 11 **E)** 4.0

```
int i=10;
double d=3.5;
long g=4;
out.print(i+d/g);
```

Question 8.

What is the output of the code segment to the right?

A) 20 12

B) 20 -8

C) 5 20

D) 5 -3

E) 7 -3

```
int m=15,n=-8;
if(m>=n)n=20;
if(n>=0)m=5;
if(n*m<=100){
  m=7;
  n=-3;}
out.print(m+" "+n);
```

Question 9.

How many asterisks are printed by the code shown to the right?

A) 4 **B)** 5 **C)** 0 **D)** 8 **E)** 9

```
int x;
for(x=4;x<=8;x++)
     out.print("*");
```

Question 10.

What is the output or the error of the code segment to the right?

A) 3

B) 5

C) Error. ArrayIndexOutOfBounds exception.

D) Error in **line #1**.

E) Error in **line #2**.

```
String []list=new String[5];//line #1
list[0]="one";
list[1]="two";
list[4]="three";
out.print(list.length);//line #2
```

Question 11.

Which of the following lines in the code segment shown on the right contains an error? All file input setup is correct.

A) line #1

B) line #2

C) line #3

D) line #4

E) None of the above. There are no errors.

```
File f=new File("datafile.dat");//line #1
Scanner scr=new Scanner(f);//line #2
while(scr.hasNextInt()) //line #3
     out.print(scr.nextInt()+" ");//line #4
```

Question 12.

What is the output of the code segment to the right?

A) 36 71

B) 49 154

C) 42 105

D) 35 70

E) 48 153

```
int x=0,y=0;
while(x<40) {
     y+=x;
     x+=7;
     }
out.print(x+" "+y);
```

Question 13.

What is the correct order of operations (from left to right) for the operators listed on the right?

A) ++ * < && +=

B) ++ && * < +=

C) += ++ && * <

D) ++ < * && +=

E) * < ++ && +=

```
&&    +=    ++    <    *
```

Question 14.

Which of the following lines of code will NOT compile?

A) `byte b=32768;`

B) `short s=32768;`

C) `int i=32768;`

D) A and B

E) B and C

Question 15.

What is the output of the code segment to the right?

A) `0 4`

B) `-9 4`

C) `14 5`

D) `-9 3`

E) `0 5`

```
ArrayList<Integer> list=new
ArrayList<Integer>();
list.add(3);list.add(14);list.add(-9);
list.set(1, 0);list.add(0, 5);
out.print(list.get(3)+" "+list.size());
```

Question 16.

What is the output of the code segment to the right?

A) `true true true`

B) `true false true`

C) `true false false`

D) `false true false`

E) `false true true`

```
String s1=new String("InvitationalA");
String s2=new String("InvitationalA");
out.print((s1==s2)+" ");
out.print(s1.equals(s2)+" ");
out.print(s2.equals("InvitationalA"));
```

Question 17.

What is the output of the code segment to the right?

A) `9.0`

B) `10`

C) `9`

D) `9.781`

E) There is no output due to an error.

```
double d1=5.031,d2=4.75;
int i=(int)d1+(int)d2;
out.print(i);
```

Question 18.

Which of the following methods will return the sum of the digits in the parameter `n`? `n` will always be positive.

A)

```
public static long sumOfDigs(long n){
  long sum=0;
  for(long x=n;x>0;x/=10)
    sum+=x/10;
  return sum;
}
```

B)

```
public static long sumOfDigs(long n) {
  long sum=0;
  String str=Long.toString(n);
  for(int i=0;i<str.length();i++)
   sum+=Long.parseLong(
        str.substring(i, i+1));
  return sum;
}
```

C)

```
public static long sumOfDigs(long n) {
  long sum=0;
  do {
    sum+=n%10;
    n=n/10;
  }while(n>0);
  return sum;
}
```

D)

```
public static long sumOfDigs(long n) {
  long sum=0;
  while(n>0) {
    n%=10;
    sum+=n;
    n/=10;
    }
  return sum;
}
```

E) More than one of the above.

Question 19.

What is the output of the code segment shown on the right?

A) `[b, tm, nsuperm, n, qu, m, n]`

B) `[b, tmansupermanaquaman]`

C) `[batmansuperm, naquaman]`

D) `[ba, tma, nsuperma, na, qua, ma, n]`

E) `[a, a, a, a, a, a]`

```
String s="batmansupermanaquaman";
String[] spl=s.split("a");
out.print(Arrays.toString(spl));
```

Question 20.

What is the output of the code segment to the right?

```
A) [8, 5, 2]     B) [9, 6, 1]     C) [8, 2, 5]
   [3, 4, 7]        [7, 4, 3]        [3, 7, 4]
   [1, 6, 9]        [2, 5, 8]        [1, 9, 6]

D) [2, 5, 8]     E) [9, 6, 1]
   [7, 4, 3]        [2, 5, 8]
   [9, 6, 1]        [7, 4, 3]
```

```
int[][] mat= {{2,5,8},
              {7,4,3},
              {9,6,1}};
int[] x=mat[0];
mat[0]=mat[2];
mat[2]=x;
for(int[] m:mat)
     out.println(Arrays.toString(m));
```

Question 21.

What is printed by the code segment listed on the right?

A) `11.25`

B) `Total=11.25`

C) `Total=11`

D) `Total=74.25`

E) `Total=74`

```
String s="Total=";
int i=7;
double d=4.25;
out.print(s+i+d);
```

Question 22.

The method shown on the right is intended to implement a binary search algorithm. Which of the following must replace **<code 1>** both places it occurs to ensure the method will compile and execute as intended?

A) `list.length/2`

B) `(front+back)/2`

C) `list.length-1/2`

D) `(front+list.length)/2`

E) `front+back/2`

Question 23.

Which of the following must replace **<code 2>** to ensure the method will compile and execute as intended?

A) `i`

B) `mid`

C) `front`

D) `back`

E) No additional code is required.

Question 24.

Once implemented correctly, and if n is the length of the array, which of the following is the strictest correct runtime?

A) O(1)

B) O(n)

C) O(n^2)

D) O(log n)

E) O(n log n)

```
//Use the following to answer questions 22,
//23 and 24.
public static int bs(int[] list,int e) {
      int i=-1;
      int front=0,back=list.length-1;
      int mid=<code 1>;
      while(back>=front) {
            if(list[mid]==e) {
                  i=mid;
                  break;
            }
            else if(e<list[mid])
                  back=mid-1;
            else
                  front=mid+1;
            mid=<code 1>;
      }
      return <code 2>;
}
```

Question 25.

What is the output of the code segment to the right?

A) `a@ $2#d*5`

B) `a w d`

C) `@ $ # *`

D) `a w 2 d 5`

E) `@ $ 2 # * 5`

```
String s="a@w$2#d*5";
Scanner scr=new Scanner(s);
scr.useDelimiter("\\w");
while(scr.hasNext())
     out.print(scr.next()+" ");
```

Question 26.

Given the declaration of `r` shown on the right, which of the following will print only values between 20 (inclusive) and 40 (exclusive)?

A) `out.print(r.nextInt(40));`

B) `out.print(r.nextInt()+20);`

C) `out.print(r.nextInt(20)+19);`

D) `out.print(r.nextInt(39));`

E) `out.print(r.nextInt(20)+20);`

```
Random r=new Random();
```

Question 27.

Which of the following could serve as a correct alternative for the condition of the while loop marked with the comment and still produce the same output?

A) `q.next()!=null`

B) `q.peek()!=null`

C) `q.hasNext()`

D) `q.size()>=0`

E) `Queue.hasNext(q)`

Question 28.

Which of the following represents the output of the code segment listed on the right before **line #1** has executed?

A) `gamma delta chi`

B) `delta gamma chi`

C) `beta chi delta gamma`

D) `alpha beta chi delta gamma`

E) `chi delta gamma`

Question 29.

Which of the following represents the output of just **line #1** in the code segment?

A) `0`

B) `3`

C) `4`

D) `5`

E) There is no output because **line #1** throws an exception

```
//Use the following code segment to
//answer questions 27, 28 and 29.
Queue<String> q=new
PriorityQueue<String>();
q.add("delta");q.add("beta");
q.add("gamma");q.add("alpha");
q.poll();q.add("chi");q.remove();
while(!q.isEmpty())//comment
     out.print(q.remove()+" ");
out.print("\n"+q.size());//line #1
```

Question 30.

Which of the following values would be returned by this call to method `mtd` shown on the right?

`mtd(9)`

A) 55

B) 34

C) 21

D) 15

E) 54

```
public static int mtd(int n) {
     if(n==0)
          return 0;
     else if(n==1)
          return 1;
     else
          return mtd(n-1)+mtd(n-2);
}
```

Question 31.

Which of the following represents the output of the `main` method shown on the right?

A) `125 5 50 10`

B) `125 55 50 10`

C) `275 5 50 10`

D) `275 55 50 5`

E) `0 50 5 10`

```
public static void main(String[] args) {
     int x=5,y=50,z=10;
     out.print(go(x,y,z)+" ");
     out.print(x+" "+y+" "+z);
}

public static int go(int x,int y,int z) {
     int p=0;
     while(x<=y) {
          p=p+x;
          x+=z;
     }
     return p;
}
```

Question 32.

The constructor for the `Contestant` class shown on the right is _________________.

A) overridden

B) overwritten

C) overloaded

D) extended

E) inherited

Question 33.

Which of the following will not correctly instantiate a `Contestant` object?

A) `Contestant c1=new Contestant("Bob","123",232);`

B) `Contestant c2=new Contestant("Sue","321");`

C) `Contestant c3=new Contestant();`

D) `Contestant c4=new Contestant("Al",231,85);`

E) None of the above. All are correct.

Question 34.

What is the output of the client code shown here?

```
Contestant c5=new Contestant();
out.print(c5.getName()+" "+c5.getIdNum()
          +" "+c5.getScore());
```

A) `0`

B) `null null 0`

C) `null 0`

D) There is no output and there are no errors.

E) There is no output due to an error.

Question 35.

Which of the following methods, when added to the `Contestant` class, will compile and correctly assign a value to the instance variable `score`?

A)
```
public int setScore(int s) {
  score = s;}
```

B)
```
public void setScore(int s) {
  return score;}
```

C)
```
public void setScore(int s) {
  s = score;}
```

D)
```
public void setScore() {
  score = s;}
```

E)
```
public void setScore(int s) {
  score = s;}
```

```
//Use the class implemented here to answer
//questions 32 - 35.
public class Contestant {

 private String name, idNum;
 private int score;

 public Contestant(String n, String id, int
 s){
  name=n;
  idNum=id;
  score=s;}

 public Contestant(String n, String id) {
  name=n;
  idNum=id;}

 public Contestant() {}

 public int getScore() {
  return score;}

 public String getName() {
  return name;}

 public String getIdNum() {
  return idNum;}
}
```

Question 36.

Which of the following represents a post-order traversal of the binary search tree illustrated on the right?

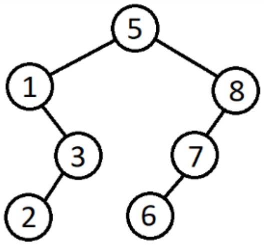
A) 5 1 3 2 8 7 6

B) 2 3 1 6 7 8 5

C) 1 2 3 5 6 7 8

D) 5 1 8 3 7 2 6

E) 8 7 6 2 3 1 5



Question 37.

Which of the following represents the longest simple cycle within the graph shown on the right?

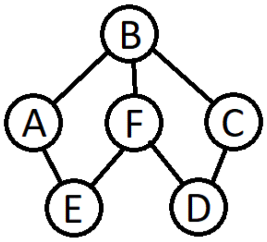
A) A B C D E A

B) A E F B A

C) C D F E A B C

D) A B C D F B C

E) A B C D F E



Question 38.

Which of the following Boolean expressions is diagrammed on the right?

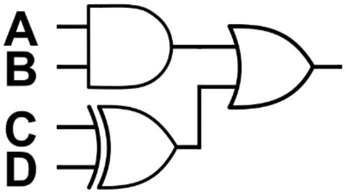
A) $A * B + C \oplus D$

B) $A + B * C \oplus D$

C) $A * B + \overline{C + D}$

D) $A + B \oplus C * D$

E) $A * B + \overline{C \oplus D}$



Question 39.

Evaluate the postfix expression shown on the right and write your answer in the blank provided on the answer document.

8 3 4 * 2 / + 5 -

Question 40.

Write the 8-bit binary two's complement representation of -90 in the blank provided on the answer document.