University Interscholastic League
Computer Science Competition

Number 148 (State - 2014)

General Directions:

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

2) **NO CALCULATOR OF ANY KIND MAY BE USED.**

3) There are 40 questions on this contest exam. 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 45 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. 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, but not on the answer sheet or Scantron card, which are 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.

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

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 an incorrect answer.

Note: Correct responses are based on Java, **J2sdk v 1.7.25**, 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

Which of these is NOT equivalent to $527_8 + 910_{10}$?

A. 1253_{10} B. 2345_8 C. $4E5_{16}$ D. 1001100101_2 E. All are equivalent

QUESTION 2

What is output by the code to the right?

A. 1.7 B. 2.4 C. 5.2

D. 5.4 E. 7.5

```
out.println(23 / 4 + 9.4 % 3);
```

QUESTION 3

What is output by the code to the right?

A. Atrue
B. true
C. falseA
D. There is no output due to a compile error.
E. There is no output due to a runtime error.

```
out.printf("%s%s",false,'A',"true");
```

QUESTION 4

What is output by the code to the right?

A. falsefalse B. falsetrue

C. truefalse D. truetrue

E. There is no output due to a compile error.

```
String s = "Tortuga";
out.print(s.contains("tor"));
out.println(s.contains("tug"));
```

QUESTION 5

What is output by the code to the right?

A. true B. false

C. There is no output due to a compile error.

D. There is no output due to a runtime error.

```
boolean p = false;
boolean q = false;
out.println(!(p^q));
```

QUESTION 6

What is output by the code to the right?

A. 14.0 B. 14

C. 15.0 D. 15

E. There is no output due to a compile error.

```
out.printf("%.1f",Math.sqrt(225));
```

QUESTION 7

What is output by the code to the right?

A. 0 88 3.0 B. 0 120 3.0

C. 2 86 3.14 D. 2 118 3.14

E. There is no output due to an error.

```
int x = 15;
int y = 'X';
double z = 3.14;
y -= x %= z;
out.println(x+" "+y+" "+z);
```

QUESTION 8

What is output by the code to the right if the values for <input1> and <input2> were "xoxoxo" and 2?

A. 2 B. 3 C. 4

D. 5 E. 7

```
String s = <input1>;
int k = <input2>;
int sum = 0;
switch(s.substring(k))
{
  case "xoxo"  : sum+=4;break;
  case "oxoxo" : sum+=3;
  case "xo"    : sum+=2;break;
  case "x"     : sum+=1;
  case "o"     : sum *= 10;
}
out.println(sum);
```

QUESTION 9

In the code to the right, what values for <input1> and <input2> would result in an output of 10?

A. "o" 0 B. "xox" 2

C. "xoxo" 0 D. None of these

E. More than one of these.

QUESTION 10

What is output by the code to the right?

A. 0 B. 1 C. 6

D. 7 E. 8

```
int j = 10000000,c=0;
do{
   j/=10; c++;
  }while(j>1);
out.println(c);
```

QUESTION 11

What is output by the code to the right?

A. 6.6 B. 8.0 C. 8.8

D. 13.2 E. 19.8

```
double [] list = {1.1,2.2,3.3};
list[1]=list[2]*2;
list[2]=list[1]*3;
out.printf("%.1f\n",list[2]);
```

QUESTION 12

Consider the data file below and code segment to the right. Assume the Scanner f object has been correctly linked to the file shown below. What is the last output of the code to the right?

```
3
The Cosmos is all that is
or ever was, or
ever will be.
```

A. The B. or

C. ever D. ever will be

E. or ever was, or

```
Scanner f = <link to data file>;
out.println(f.nextInt());
out.println(f.nextLine());
out.println(f.nextLine());
out.println(f.next());
```

QUESTION 13

What is output by the code to the right?

A. 0 B. 1

C. 2 D. 3

E. 4

```
double x = 0.0; int y = 0;
double z = Math.toRadians(360);
do{
     x+=Math.PI;
     y++;
}while(x<=z);
out.println(y);
```

QUESTION 14

What is output by the code to the right?

A. true B. false

C. There is no output due to a compile error.

D. There is no output due to a runtime error.

```
boolean b = true && false | true;
out.println(b);
```

QUESTION 15

What is output by the code to the right?

A. 4 B. 8 C. 16

D. 32 E. 64

```
out.println(Double.SIZE);
```

QUESTION 16

What is output by the code to the right?

A. TomDickHarry

B. TomDickLarry

C. DickHarryHarry

D. DickHarryLarry

E. TomMoeLarry

```
ArrayList<String> list = new
 ArrayList<String>();
list.add("Tom");
list.add("Dick");
list.add("Harry");
list.add("Larry");
list.add("Moe");
list.add("Curly");
out.print(list.get(1));
Collections.sort(list);
out.print(list.get(2));
Collections.reverse(list);
out.println(list.get(3));
```

QUESTION 17

Question Omitted

QUESTION 18

Which of the following is NOT an output of the code segment to the right?

A. 0000 B. 0100 C. 0110

D. 1100 E. 1110

```
for(int p = 0;p <= 1; p++)
 for(int q = 0;q <= 1; q++)
  for(int r = 0;r <= 1; r++)
  {
      boolean P = p==1;
      boolean Q = q==1;
      boolean R = r==1;
      boolean S = (P|Q)&(P&!R);
      int s = S?1:0;
      out.print(""+p+q+r+s+" ");
  }
```

QUESTION 19

What is output by the code to the right?

A. 42 B. 126 C. 976

D. 1976 E. 2014

```
int x = <the year of this UIL test>;
int y = <# of pounds in one ton>;
int z = <square ft in a square yd>;
out.println(x%y*z);
```

QUESTION 20

What is output by the code to the right?

A. 67 B. 604 C. 1611

D. 3021 E. 4028

```
int y = 2014;
out.println(y<<3>>>2<<4/10);
```

QUESTION 21

What is output by the `client code` to the right?

A. 160.00 B. 400.00 C. 520.00

D. 560.00 E. 590.00

```
static double myst(double a,double b)
{
  double c = 0;
  if(a>48.0)
     {c+=(a-48)*2*b; a=48;}
  if(a>40.0)
     {c+=(a-40)*3/2*b; a=40;}
  c+=a*b;
  return c;
}
//client code
out.printf("%.2f\n",myst(50,10));
```

QUESTION 22

What is output by the code to the right?

A. Iwttowns! B. IIwttowns!

C. Iwttownse D. IIwttownse

E. There is no output due to an error.

```
String s = "I want to win state!";
String [] ss = s.split(" ");
String w = "";
for(String b:ss)
{
  char [] list = b.toCharArray();
  w+=""+list[0]+list[list.length-1];
}
out.println(w);
```

QUESTION 23

What is output by the code to the right?

A. truetruetrue

B. truefalsetrue

C. truetruefalse

D. falsefalsetrue

E. falsefalsefalse

```
String s = "1a2b3c4d5e";
boolean p = s.matches(".*\\d\\w.+");
boolean q = s.matches(".\\D\\S.*");
boolean r = s.matches("[abc]+");
  out.println(""+p+q+r);
```

QUESTION 24

What is output by `statement 1` in the client code to the right?

A. 3

B. 4

C. 5

D. 6

E. 7

QUESTION 25

What is output by `statement 2` in the client code to the right?

A. 5

B. 6

C. 7

D. 8

E. 9

```
static int A(int m, int n)
{
  if(m==0)
     return n+1;
  if(m!=0&&n==0)
     return A(m-1,1);
  if(m!=0&&n!=0)
     return A(m-1,A(m,n-1));
  return 0;
}
//statement 1
out.println(A(1,3));
//statement 2
out.println(A(2,3));
```

QUESTION 26

Which of the following concepts is NOT represented by the code to the right?

A. inheritance
B. polymorphism
C. overloading
D. overriding
E. All are represented

QUESTION 27

Which of these best replaces **<statement 1>** in the code to the right?

A. `Comparable o`
B. `Object o`
C. `Ork o`
D. `Mork o`

QUESTION 28

What is output by **`segment one`** in the client code to the right?

A. `Ork 0 3 Mork -1 4 Mork -1 0`
B. `Ork 0 3 Mork -1 4 Ork -1 0`
C. `Ork 0 3 Mork -1 4 Mork -1 4`
D. `Ork 0 3 Mork -1 4 Ork -1 4`
E. There is no output due to an error.

QUESTION 29

What is output by **`segment two`** in the client code to the right?

A. `0 0 0`
B. `1 1 1`
C. `-1 0 -1`
D. `-1 0 1`
E. `1 0 -1`

```
public class Ork implements
     Comparable<Ork>{
  int snark,shazbat,nanu;
public Ork(){}
public Ork(int n, int s, int u){
  snark=n;shazbat=s;nanu=u;
}
public String toString(){
  return "Ork "+(snark+shazbat-nanu);
  }
public int compareTo(<statement 1>){
  int x = snark+shazbat-nanu;
  int y = o.snark+o.shazbat-o.nanu;
  return x>y?1:x<y?-1:0;
}
}
class Mork extends Ork
{
  int nanu;
public Mork(){}
public Mork(int n, int s,
            int u, int a)
{
  snark=n;shazbat=s;nanu=u;
  this.nanu=a;
}
public String toString()
{
  return "Mork "+(snark+shazbat-
nanu);
}
}
//client code
  Ork one = new Ork(1,2,3);
  Mork two = new Mork(1,2,3,4);
  Ork trey = new Mork(1,2,3,4);
//segment one
  out.print(one+" "+one.nanu+" ");
  out.print(two+" "+two.nanu+" ");
  out.println(trey+" "+trey.nanu);
//segment two
  out.print(one.compareTo(two)+" ");
  out.print(trey.compareTo(two)+" ");
  out.println(two.compareTo(one));
```

QUESTION 30

What is output by the code to the right?

A. `000000000000000` (15 zeroes)
B. `111111111111111` (15 ones)
C. `00000000000000000111111111111111` (17 0s, 15 1s)
D. `11111111111111110000000000000000` (17 1s, 15 0s)
E. There is no output due to an error.

```
short s = Short.MAX_VALUE;
String t = Integer.toBinaryString(s);
out.println(t);
```

QUESTION 31

Which of the following correctly replaces <value> in the code to the right in order to output the value 1?

A. 3 B. 8 C. 9

D. 10 E. 1000

```
int x = <value>;
out.println(1000>>>x);
```

QUESTION 32

Which of the following represents the missing lines **<?>** in the output shown in the code to the right?

A.

```
0x1.cp1
0x1.0p2
0x1.4p2
```

B.

```
0x1.0p2
0x1.4p2
0x1.8p2
```

C.

```
0x1.fp1
0x1.5p2
0x1.9p2
```

D.

```
0x1.0p2
0x1.2p2
0x1.4p2
```

E.

```
0x1.10p2
0x1.12p2
0x1.14p2
```

```
double d = 1.0;
while(d<11.0)
out.println(Double.toHexString(d++));

//partial output
0x1.0p0
0x1.0p1
0x1.8p1
<?>
<?>
<?>
0x1.cp2
0x1.0p3
0x1.2p3
0x1.4p3
```

QUESTION 33

What is output by the code to the right?

A. 18 B. 19 C. 21

D. 24 E. There is no output due to an error.

```
int [] list = new int[10];
Arrays.fill(list,1,10,1);
Arrays.fill(list,2,9,2);
Arrays.fill(list,3,8,3);
int sum=0;
for(int x:list)
 sum+=x;
out.println(sum);
```

QUESTION 34

What is output by the code to the right?

A. This tess is ss eass.

B. This tesst is sso eassy.

C. Thisstess issss eass.

D. Thiss tesst iss sso eassy.

E. There is no output due to an error.

```
String s = "This test is so easy.";
String t = s.replaceAll("s\\w","ss");
out.println(t);
```

QUESTION 35

In the chart to the right, representing the most restrictive bound on the runtime of each process in each scenario, where N represents the number of items in list, how many scenarios have a runtime of O(N)?

A. 0
B. 2
C. 6
D. 8
E. 10

Algorithm	Scenarios/Big O Time Complexity		
	Best	Average	Worst
Quicksort	?	?	?
Mergesort	?	?	?
Heapsort	?	?	?
Bubble Sort	?	?	?
Insertion Sort	?	?	?
Selection Sort	?	?	?

QUESTION 36

Using the same chart, how many scenarios have a runtime of O(N^2)?

A. 6
B. 7
C. 8
D. 9
E. 10

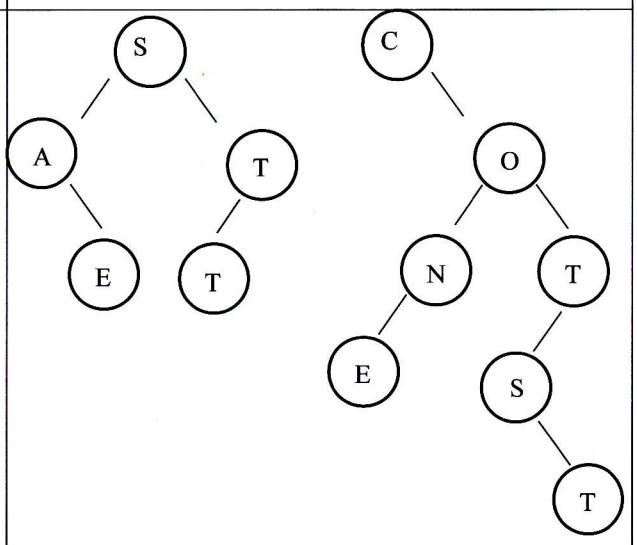
QUESTION 37

To the right is a graph made up of two binary search trees for the strings STATE and CONTEST.

The internal path length of the STATE tree is 6, which means that the total number of steps from each non-root node back to the root is 6. The A and T nodes are each 1 step away, and the E and T nodes are each 2 steps away, for a total of 6 steps.

What is the internal path length of the CONTEST tree ?

A. 6 B. 10 C. 12
D. 15 E. 22



QUESTION 38

How many nodes in this graph (both trees) have only one child?

A. 5 B. 6 C. 7
D. 8 E. 9

QUESTION 39

After the push and pop sequence shown on the right involving two parallel stacks, where the first argument of each command corresponds with the first stack, and the second argument to the second stack, which value would be the next one popped from the second stack?

A. 1 B. 2
C. 3 D. 6
E. 9

```
Push 4 5
Push 1 2
Push 6 3
Pop x y
Push 9 7
Pop x y
Push 5 8
Pop x y
```

QUESTION 40

In a directed graph such as the one on the right, there are often simple paths (no repeated nodes) that form a cycle (back to the starting node), such as these two examples, CGC (also named GCG) and ABCGDA (also named BCGDAB and CGDABC). How many unique cycles are there in this graph?

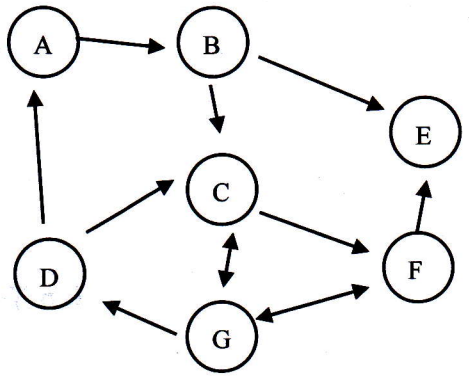
A. 4

B. 5

C. 6

D. 7

E. 8



Computer Science Answer Key
UIL State 2014

1)	D	11)	E	21)	D	31)	C
2)	D	12)	B	22)	B	32)	B
3)	C	13)	D	23)	C	33)	C
4)	B	14)	A	24)	C	34)	A
5)	A	15)	E	25)	E	35)	B
6)	C	16)	C	26)	E	36)	C
7)	C	17)	omitted	27)	C	37)	D
8)	C	18)	D	28)	A	38)	B
9)	B	19)	B	29)	D	39)	B
10)	D	20)	E	30)	B	40)	D

Note to Graders:

- All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g. error is an answer). **Ignore any typographical errors.**
- Any necessary Standard Java 2 Packages are assumed to have been imported as needed.
- Assume any undefined (undeclared) variables have been defined as used.

Explanations:

1. $527_8 + 910_{10} = 1253_{10} = 2345_8 = 4E5_{16} = 10011100101_2$
2. 23 / 4 + 9.4 % 3 = 5 + 0.4 = 5.4
3. The %s format specifier of the printf statement accepts the string representation of any data element, and any extra parameters, such as the "true" string in this statement, are ignored.
4. The contains method of the String class returns true if there is an exact match of the given parameter somewhere in the string. "tor" does not match due to the uppercase "T", but "tug" does.
5. This expression, NOT(P XOR Q), simplifies to be P equals Q, which means that for the expression to evaluate to true, P and Q must be the same value, either both false or both true.
6. The square root of 225 is 15, and since the Math.sqrt method returns a double, and the %.1f format specifier is used, the output is 15.0.
7. A compound expression evaluates from right to left, therefore x changes from 15 to 2 (15 % 3.14 = 2.44, which is autocast to 2), y gets the ASCII value of 'X' (88), which is then subtracted by 2, resulting in 86. The value of z does not change.
8. The String produced by `"xoxoxo".substring(2)` is `"xoxo"`, which matches the first case in the switch, resulting in a value of 4 for sum.
9. The only input values that produce the value 10 are "xox" and 2, resulting in the String "x", which matches the fourth case, adds 1 to sum, then drops down to the next case and multiplies sum by 10, resulting in an output value of 10.
10. This is essentially the calculation of the log, base 10, of 10,000,000, which is 7. This means that 10,000,000 is equal to 1 X 10^7.
11. The sequence of calculations is as follows:
    - list[1] (currently 2.2) is assigned the value 3.3*2, which is 6.6
    - list[2] (currently 3.3) is assigned the value 6.6*3, which is 19.8
12. The input sequence is as follows:
    - `nextInt` grabs the 3
    - `nextLine` gobbles up the whitespace after the 3
    - `nextLine` takes the String, `"The Cosmos is all that is"`
    - `next` gets the String, `"or"`, the last output
13. Since `Math.toRadians(360)` produces the value 2PI, it will take the variable x three additions of PI to exceed that value, therefore the value of y after the loop is 3.
14. Even though this expression is using the `&&` logic operator and the `|` bitwise operator (which really evaluates first in the Java order of precedence), the result is the same…true. False OR True is True, and then True AND True is True.
15. The double data type uses 64 bits of storage.
16. After all of the elements are added, "Dick" is in position 1. Once the list is sorted, "Harry" is in position 2, and after the list is reversed, "Harry" is in position 3.
17. Omitted
18. The output of 1100 should be 1101, which shows that when P = true, Q = true, and R = false, (P OR Q) AND (P AND NOT R) evaluates to `(true OR true)AND(true AND NOT false)`, which simplifies to `true AND (true AND true)` which further simplifies to `true`, not `false` as 1100 shows.
19. With the values 2014 (year of this test), 2000 (pounds in one ton), and 9(square feet in one square yard), the expression is 2014%2000*9, which evaluates to 14*9, or 126. Mod has the same level of precedence as times, therefore it occurs first and is evaluated first.
20. Since division has a higher precedence than shift operations, 4/10 evaluates to 0, which causes y to have a net left shift of 1, which means 2014 is multiplied by 2^1, resulting in 4028. The most likely error of 1611 would be the result if the shift operations went first (net left shift 8), with the shift result of 16112 integer divided by 10 to make 1611.
21. This method calculates a person's paycheck, with double overtime for any hours over 48, time-and-a-half for hours over 40 up to 48, and regular pay for 40 hours or less. For 50 hours at $10 an hour, the pay is calculated at 2 hours at double time (2*2*10), plus 8 hours at time-and-a-half (8*1.5*10), plus 40 hours at regular pay (40*10), for a total paycheck of $560.00.
22. After the initial " " split into the array of Strings, the for loop takes the first and last character of each word and builds a new String. `"I want to win state!"` produces `"IIwttowns!"`. The "I" counts twice since it is both the first letter and last letter of that word.
23. The first match `(".*\\d\\w.+")` looks for zero or more characters + a single digit + single word character + one or more of any character, which is `true` since the ".*" can be ignored, the "1" matches the single digit, the "a" matches the single word character, and the rest of the string matches the ".+". The next match `(".\\D\\S.*")` means a single character + single non-digit + single non-space + zero or more characters, also `true`. The third match `("[abc]+")` checks for one or more characters from the `[abc]` set, and nothing more. This match is false since the first character is NOT from the `[abc]` set.
24. In computability theory, the **Ackermann function**, named after Wilhelm Ackermann, is one of the simplest and earliest-discovered examples of a total computable function that is not primitive recursive. All primitive recursive functions are total and computable, but the Ackermann function illustrates that not all total computable functions are primitive recursive. See the recursive trace below. As you can see, it doesn't take much for this recursive process to get out of hand. As an example, A(4,1) will most likely cause a stack overflow if you try to run it on a PC, and just forget about even trying to trace it by hand!
25. Same as 24.
26. All of these are classic Object Oriented Programming concepts. Inheritance is represented by the Mork class extending the Ork class. Polymorphism is shown by the fact that the toString method defined in both Ork and Mork classes. This is also referred to as overriding. Overloading is represented by two constructors in either class with different parameter signatures.
27. Since the Comparable is implemented specifically for the Ork class, the parameter must be of type Ork, and not just Object.
28. Since the output of an object is controlled by the toString method, examine the toString method for each class and you will see the result. Also, since the nanu field is present in both classes, the toString method will use the nanu version that belongs to the object, but when outputting the nanu field directly, the one that belongs to the object reference is used. However, in the trey version, a curious thing happens – the compiler uses the value zero instead of the super class nanu value…interesting.
29. In the compareTo method arithmetic, the `one` object evaluates to 0 (1+2-3), and both the `two` and `trey` objects evaluate to 3 (1+2-0). Therefore the `one` object is less than both `two` and `trey` objects, and the `two` and `trey` objects are equal to each other.

30. The maximum value of the 16-bit short data type is 32767, which is 0111111111111111 in binary (0 + 15 1s), and is output without the leading zero.
31. The integer value 1000 takes 9 divisions by 2 to reach a value of 1….500, 250, 125, 62, 31, 15, 7, 3, 1.
32. See the Double class in the Java API for further clarification of the Double.toHexString method.
33. The contents of the list after the three fill statements is [0, 1, 2, 3, 3, 3, 3, 3, 2, 1], with a sum of 21.
34. In this replacement process, any "s" followed by another word character is replaced with "ss".
35. The Bubble and Insertion sorts each have an O(N) time complexity in the best case scenario, which is when the list is already sorted, or very nearly sorted.
36. Eight scenarios are rated at O(N^2) – Quicksort(worst), Bubble and Insertion(average and worst), Selection(all three scenarios)
37. O is 1 step away, N and T 2 steps, E and S 3 steps, and T 4 steps, for a total of 1+2+2+3+3+4 = 15
38. In the STATE tree, only A and T on level 1 have just one child, and in the CONTEST tree, C, N, T, and S have just one child, for a total of 6 nodes with just one child.
39. The remaining values in the first stack after the sequence (top to bottom) are 1 and 4, with 2 and 5 left in the second stack, 2 being the next one to be popped from the second stack.
40. The seven cycles in this directed graph are: GFG, GCG, CGDC, CFGC, CFGDC, ABCGDA, and ABCFGDA