

Clements High School

Computer Science Competition 2014

General Directions (Please read carefully!):

- 1) DO NOT OPEN EXAM UNTIL TOLD TO DO SO.
- 2) **NO CALCULATORS 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. **All provided code segments are intended to be syntactically correct, unless otherwise stated. Ignore any typographical errors and assume any undefined variables are defined as used.**
- 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.
- 10) Assume that any necessary import statements for standard Java packages and classes (e.g. `.util`, `ArrayList`, etc.) are included in any programs or code segments that refer to methods from these classes and 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 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

What is 101101_2 plus $1E_{16}$?

- A. 75_{10} B. $5A_{16}$ C. $4C_{16}$ D. 85_{10} E. 101011_2

QUESTION 2

What is output by the code to the right?

- A. 2 B. 10 C. 12
D. 11 E. There is no output due to a run-time error

```
int x = 3, y = 6;
int z = x + y * y / x - x;
out.println(z);
```

QUESTION 3

What is output by the code to the right?

- A. 15
 Good job!!!!
B. 10
 Good job!!
C. 10
 Good job!!!!
D. 15
 Good Job!!!!
E. 15
 Bad job!!!!

```
int a[] = {-1, 1, -2, 8, 4};
int sum = 0;
int i = 0;
for(; i < 5; i++) {
    sum += a[i];
}
out.println(sum + i);
out.println("Good job!!!!");
```

QUESTION 4

What is output by the code to the right?

- A. Get me to the school on time
B. Get you to the school on time
C. Get me to the school on tiyou
D. Get you to the school on tiyou
E. There is no output due to a run-time error

```
String x;
x = "Get me to the school on time";
x.replace("me", "you");
out.println(x);
```

QUESTION 5

What values for a, b, and c make the output to the right false?

- A. only b must be false
B. either a and b or b and c must be false
C. a, b, and c must all be false
D. b must be true
E. a and c must be true

```
boolean a, b, c, d;
d = a || b && b || c;
out.println(d);
```

QUESTION 6	What is the range of the output by the code to the right? A. [-10,10) B. [0,24) C. [5,14) D. [5,10) E. [-10,14)	<pre>int x = (int)(Math.random()*5+5); int y = (int)(Math.random()*24-10); out.println(Math.max(x,y));</pre>
QUESTION 7	What is output by the code to the right? A. 2 B. 4 C. 3 D. 19 E. 5	<pre>int a; int x = 6; int y = 10; int z = 8; a = x + y + z % 5; out.println(a);</pre>
QUESTION 8	What is output by the code to the right? A. 112 B. 37 C. 87 D. 62 E. 137	<pre>int x = 37; if(x<25) x+=25; if(x<50) x+=25; if(x<75) x+=25; if(x<100) x+=25; out.println(x);</pre>
QUESTION 9	What is output by the code to the right? A. KS AOOA S B. KSCL ARYO C. SKLCA YRO D. YOL_KK_LO E. SKA OO AS	<pre>String s = "KC ROYALS"; char let; int len = s.length(); for(int i=0; i<len; i++) { if((i+5)%2==0) let=s.charAt((len-i)%s.length()); else let=s.charAt((len+i)%s.length()); out.print(let); } out.println();</pre>
QUESTION 10	What is output by the code to the right? A. 4 B. 3 C. 0 D. 8 E. There is no output due to a run-time error	<pre>int[] list = {2,3,8,1,4,0,3,2}; out.println(list[list[5]]);</pre>

QUESTION 11

Which of the following correctly instantiates a Scanner object?

- A. Scanner input = new Scanner(System.out);
- B. Scanner input = new String("Bobby");
- C. Scanner input = System.in;
- D. Scanner input = new Scanner("Bobby");
- E. Scanner input = new Scanner();

QUESTION 12

What is output by the code to the right?

- A. 17
- B. 15
- C. 16
- D. 56
- E. 50

```
int x = 1;
for(x = 2; x < 15; x+=2)
    x++;
out.println(x);
```

QUESTION 13

What is the output for the code to the right? A '-' represents a space.

- A. 2.500
- B. -----2.50
- C. -2.50
- D. --2.5
- E. 2.50

```
System.out.printf("%5.2f", 2.5);
```

QUESTION 14

What is output by the code to the right?

- A. 2147483647
- B. 65534
- C. 32767
- D. 4294967294
- E. 9223372036854775807

```
out.println(Integer.MAX_VALUE);
```

QUESTION 15

What is output by the code to the right?

- A. [Pat, Mat, Sat, Cat]
- B. [Pat, Mat, Cat, Fat]
- C. [Pat, Sat, Cat, Fat]
- D. [Fat, Mat, Cat, Pat]
- E. There is no output due to a run-time error

```
ArrayList<String> list =
    new ArrayList<>();
list.add("Pat");
list.add("Mat");
list.add("Sat");
list.add("Cat");
list.remove(2);
list.add(1,"Fat");
list.add(list.remove(0));
out.println(list);
```

QUESTION 16

Which of the following correctly places a modifier method at <*1> in the code to the right?

- A. fight += f; }
- B. public void add(int x){ fight += x; }
- C. public add(int x){ fight += x; }
- D. public GoTA(int x){ fight = x; }
- E. public int add(int x){ fight += x; }

QUESTION 17

Which of the following correctly places an accessor method at <*2> in the code to the right?

- A. public int getMoney(){out.print(money);}
- B. public int getMoney(int x){return x;}
- C. public void getMoney(){return money;}
- D. public void getMoney(int x){return money;}
- E. public int getMoney(){return money;}

QUESTION 18

What is output by line <*1> in the code on the right?

- A. 0
- B. 2
- C. 8
- D. 4
- E. 5

QUESTION 19

What is output by line <*2> in the code to the right?

- A. 16
- B. 20
- C. 36
- D. 24
- E. There is no output due to a run-time error

QUESTION 20

What is output by the code to the right?

- A. 10
- B. 0
- C. 28
- D. 1
- E. 20

```
class GoTA
{
    private int fight, money, spy;
    public GoTA(int f, int m, int s)
    {
        fight = f;
        money = m;
        spy = s;
    }
    // <*1> modifier method
    // <*2> accessor method
}
```

```
int[][] mat = {{6,4,6,2},
               {8,2,7,4},
               {5,0,1,7},
               {4,2,2,9},
               {1,3,1,4}};

out.println(mat[2][1]); //<*1>
int k = 4;
int count = 0;
for(int i=0; i<mat.length; i++)
    for(int j=0; j<mat[i].length; j++)
    {
        count++;
        if(mat[i][j]==k)
            mat[i][j--]--;
    }
out.println(count); //<*2>
```

```
int x = 17;
int y = 20;
int z = 47;
int a = y & z>>1 | x & y>>1;
out.println(a);
```

QUESTION 21

Which of the following correctly replaces `<*1>` in the code to the right?

- A. `r<3 && c<2`
- B. `r==c && c>=2`
- C. `r>=3 && c>=2`
- D. `r>=3 || c>=2`
- E. `r>=3 && c<2`

QUESTION 22

What would be stored in `mat[0][0]` after a call to `mys1` with the following matrix?

```
int mat[][] = {{1,3,3,3,5},  
               {5,5,3,9,4},  
               {4,7,2,2,4},  
               {3,0,4,4,2},  
               {9,1,0,3,4}};
```

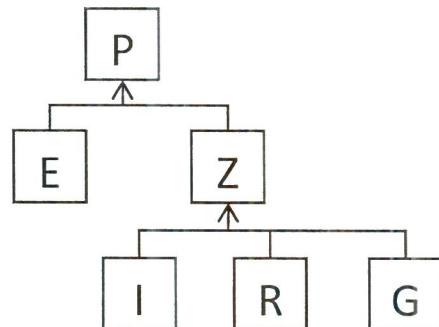
- A. 4
- B. 1
- C. 5
- D. 3
- E. -1

QUESTION 23

What can be discerned from the basic UML inheritance diagram to the right?

- A. P is a Z
- B. E is a I
- C. I has a P
- D. R is a P
- E. more than one of these are correct

```
public static void mys1(int[][] mat)  
{  
    for(int r=0; r<mat.length; r++)  
        for(int c=0; c<mat[r].length; c++)  
            if( //<*1> )  
                mat[r-3][c-2] += mat[r][c];  
            else  
                mat[r][c] --;  
}
```



QUESTION 24

What is output by line **<*1>** in the code to the right?

- A. 36
- B. 50
- C. 5
- D. 45
- E. 14

QUESTION 25

What is output by line **<*2>** in the code to the right?

- A. 52
52
- B. 22
44
- C. 7
22
- D. 7
52
- E. 52
22

QUESTION 26

What is output by line **<*3>** in the code to the right?

- A. 163
- B. 164
- C. 61
- D. 64
- E. 169

```

class W
{
    private int x;
    public W(int k){x = k;}
    public void m(int k){x+=k;}
    public int get(){return x;}
    public void p(){out.println(x);}
}

class X extends W
{
    private int x;
    public X(int k){
        super(5);
        x = k;
    }
}

class Y extends W
{
    private int x;
    public Y(int k){
        super(7);
        x = 2*k;
    }
    public void m(int k, int j){
        x+=k-j;
    }
    public void p(){
        super.p();
        out.println(x);
    }
}

//////////////CLIENT CODE///////////////
X t = new X(36);
t.m(9);
t.p();      //<*1>
Y k = new Y(22);
k.m(19,11);
k.p();      //<*2>
W[] list = new W[5];
list[0] = new W(37);
list[1] = new X(34);
list[2] = new Y(38);
list[3] = new Y(27);
list[4] = new X(28);
int sum = 0;
for(int i=0; i<list.length; i++)
    sum+=list[i].get();
    out.println(sum);    //<*3>

```

QUESTION 27

What is output by the code on the right?

- A. [46, 26, 18, 15, 9, 38, 12, 45]
- B. [9, 12, 15, 18, 26, 38, 45, 46]
- C. [9, 15, 12, 45, 18, 38, 26, 46]
- D. [9, 15, 45, 46, 12, 18, 38, 26]
- E. [45, 12, 38, 9, 15, 18, 26, 46]

```
PriorityQueue<Integer> x;
x = new PriorityQueue<>();
x.add(46);
x.add(26);
x.add(18);
x.add(15);
x.add(9);
x.add(38);
x.add(12);
x.add(45);
out.print(x);
```

QUESTION 28

What is returned by the method call `mys2 ("abcde")`?

- A. abcde
- B. ababcaabcd
- C. cdedee
- D. cddee
- E. infinite recursion

```
public static String mys2(String x)
{
    char s = x.charAt(0);
    String y =
        x.substring((s+5)%x.length());
    if(x.equals(y))
        return y;
    return y+mys2(y);
}
```

QUESTION 29

What is returned by the method call `mys2 ("sturges")`?

- A. sturgessturgesrgesrges
- B. turgesrgesrges
- C. sststurgsturges
- D. sturges
- E. turgesurgesgesges

QUESTION 30

Which of the following correctly replaces `<*1>` in the code to the right such that the value 211 is outputed?

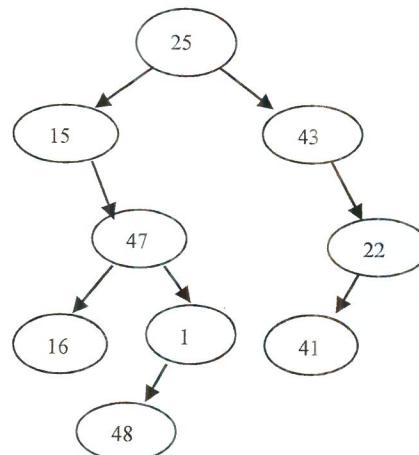
- A. 40
- B. 131
- C. 82
- D. 46
- E. 26

```
x = <*1>;
System.out.println(x^145|x);
```

QUESTION 31

What would be printed out if an pre-order traversal is used on the tree to the right?

- A. 16 48 1 47 15 41 22 43 25
- B. 25 15 47 16 1 48 43 22 41
- C. 25 15 43 47 22 16 1 41 48
- D. 1 15 16 22 25 41 43 47 48
- E. 15 16 47 48 1 25 43 41 22

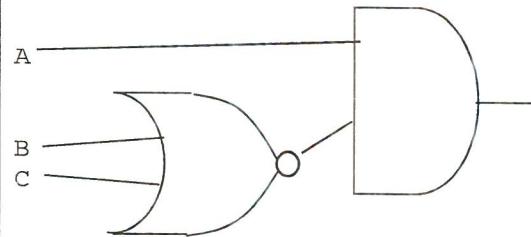


QUESTION 32 <p>What is output by the code on the right?</p> <p>A. [2,12,15,17,19,35] B. [2,2,12,19,35,17,15] C. [2,15,35,19,17,12] D. [2,12,19,35,17,15] E. [2,2,12,15,17,19,35]</p>	<pre>Stack<Integer> st = new Stack<>(); ArrayList<Integer> list; list = new ArrayList<>(); int[] x = {2,15,35,19,17,12}; for(int i=0; i<x.length; i++) { if(st.isEmpty()) st.push(x[i]); while(!st.isEmpty() && st.peek()>x[i]) list.add(st.pop()); st.push(x[i]); while(!list.isEmpty()) st.push(list.remove(0)); } out.println(st);</pre>
QUESTION 33 <p>Which of the following correctly replaces <*1> in the code to the right such that x is split at the pipes () ?</p> <p>A. x.split("\\\\ "); B. x.split(" "); C. x.split("^ [a..z]"); D. x.split("[]"); E. x.split("\ ");</p>	
QUESTION 34 <p>What is output by the code on the right?</p> <p>A. [DAL, DEN, HEL, KC, LA, LV, P, SEA, SF, SL] B. [P, SL, HEL, DEN, LV, LA, SF, SEA, KC, DAL] C. [P, SL, HEL, DEN, SF, LA, LV] D. [P, SF, SL, HEL, DEN, LV, LA] E. [DEN, HEL, LA, LV, P, SF, SL]</p>	<pre>String x; x="P SF SL HEL DEN LV LA SEA DAL KC"; String[] c = //<*1></pre>
QUESTION 35 <p>Assume the code to the right was executed with a directed graph structure instead of a map. How many edges would the graph contain?</p> <p>A. 13 B. 18 C. 7 D. 9 E. 10</p>	<pre>Map<String, String> map; map = new TreeMap<String, String>(); map.put(c[0], c[1]); map.put(c[0], c[2]); map.put(c[2], c[0]); map.put(c[2], c[3]); map.put(c[3], c[4]); map.put(c[4], c[2]); map.put(c[2], c[1]); map.put(c[1], c[2]); map.put(c[1], c[6]); map.put(c[6], c[1]); map.put(c[6], c[5]); map.put(c[5], c[2]); map.put(c[5], c[6]); out.println(map.keySet());</pre>
QUESTION 36 <p>Which of the following is a simple path (repeats no vertices) based on the graph created by the code to the right?</p> <p>A. LA LV SL P SF B. SL HEL DEN SL C. LA LV SL DEN P SF D. LA LV SL P SL SF E. LA LV SF</p>	

QUESTION 37

Which triplet (A,B,C) will make the circuit to the right true?

- A. (false, false, false)
- B. (true, true, true)
- C. (true, true, false)
- D. (false, true, true)
- E. (true, false, false)



QUESTION 38

Convert the infix notation equation to the right into a prefix notation equation.

- A. EAAF+B-*DB+*CB-/-
- B. -E/**A-+AFB+DB-CB
- C. -E**/A+-AFB+DB-CB
- D. EAAFB--*DB+*CB/- -
- E. -EA*+-AFB*+DB/-CB

$$E-A*(A+F-B)*(D+B)/(C-B)$$

QUESTION 39

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.

$$A*B*C+A*B$$

Simplify the Boolean algebra expression.

QUESTION 40

OPEN ENDED QUESTION – Fill in the blank spaces with the proper bits (1 or 0) 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 binary representation of x?

bit bit bit bit bit bit bit bit

--	--	--	--	--	--	--	--

byte x = (byte) (-56);

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.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, int 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()

class java.util.ArrayList<E> implements List<E>
    Methods in addition to the List methods:
    ○ 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.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.util.Stack<E>
  o boolean isEmpty()
  o E peek()
  o E pop()
  o E push(E item)

interface java.util.Queue<E>
  o boolean add(E e)
  o boolean isEmpty()
  o E peek()
  o E remove()

class java.util.PriorityQueue<E>
  o boolean add(E e)
  o boolean isEmpty()
  o E peek()
  o E remove()

interface java.util.Set<E>
  o boolean add(E e)
  o boolean contains(Object obj)
  o boolean remove(Object obj)
  o int size()
  o Iterator<E> iterator()
  o boolean addAll(Collection<?> extends E> c)
  o boolean removeAll(Collection<?> c)
  o 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>
  o Object put(K key, V value)
  o V get(Object key)
  o boolean containsKey(Object key)
  o int size()
  o Set<K> keySet()
  o 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>
  o K getKey()
  o V getValue()
  o V setValue(V value)

interface java.util.Iterator<E>
  o boolean hasNext()
  o E next()
  o void remove()

interface java.util.ListIterator<E> extends
java.util.Iterator<E>
  Methods in addition to the Iterator methods:
  o void add(E e)
  o void set(E e)

```

Clements H.S
Computer Science Contest Key

November 15, 2014

1) A	21) C
2) C	22) A
3) A	23) D
4) A	24) E
5) C	25) D
6) C	26) C
7) D	27) C
8) A	28) D
9) A	29) E
10) D	30) C
11) D	31) B
12) A	32) B
13) C	33) D
14) A	34) E
15) D	35) A
16) B	36) A
17) E	37) E
18) A	38) B
19) D	39) AB or A&&B
20) E	40) 1100 1000

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.

Brief Explanations:

1. $101101_2 + 1E_{16} \rightarrow 101101 + 11110 \rightarrow 1001011 \rightarrow 64+8+2+1 \rightarrow 75_{10}$
2. $3+6*6/3-3 \rightarrow 3+36/3-3 \rightarrow 3+12-3 \rightarrow 12$
- 3.
4. Strings are immutable, so there is no change to the String
5. the `&&` occurs before the `||`, hence you are dealing with `A||B||C`, which means all have to be false
6. The range would be from the largest low end of each variable to the largest high end of each variable. Arguably, there will be more numbers near 5 than there would be near 14.
7. $6+10+8\%5 \rightarrow 6+10+3 \rightarrow 19$
8. There is no else statements and the increase always lifts it to the next level, so 75 is added to 37.
9. The if statement will rotate from $+0, -1, +2, -3, +4, -5, +6, -7, +8, -9$
10. `list[5]->0, list[0]->2, list[2]->8`.
11. Scanner can connect a stream to a String, it will only read a stream.
12. This accumulation is actually changing the iterator which will be 17 because `x` at 14 becomes 17.
13. `%5.2f` first rounds the corresponding float value, 2.5, to the nearest hundredth (due to the .2), so the 2.5 value is formatted to 2.50. The 5 adds spaces to the left of the value until the string becomes 5 characters in length - in this case, `printf` adds one space. The answer is therefore (C), which is -2.50 (the dash represents a space, as stated in the problem).
14. an Integer's max value is $2^{31}-1$ which is roughly 2 billion and change. This is because an Integer is a signed int and half of the values must be negatives and zero is counted as a positive number.
15. The `<>` in the instantiation is fine, Java 7 allows you to do this.
16. a modifier method should be public, can be a return type (but you better do a return).
17. the answer B is not an accessor method because at no time does it access a private member of the class.
18. the rows and the columns start at index 0, so it is the 3rd row and the 2nd column
19. There are four 4s that will cause the `j` to do an extra step, so it is 20 steps + 4 steps = 24 steps. The postfix on `j` does not cause an index out of bounds error.
20. $20 \& 47 >> 1 \mid 17 \& 20 >> 1$. `>>` means divide by 2^x . So, $20 \& 47 / 2 \mid 17 \& 20 / 2 = 20 \& 23 \mid 17 \& 10 = 10100 \& 10111 \mid 10001 \& 01010 = 10100 \mid 00000 = 101000 = 20$.
21. looking at the true conditional branch, you must make `r>=3` and `c >= 2` in order not to go out of bounds.
22. `mat[0][0]` will first decrease to 0 when it is first used, and then changed to `mat[3][2]`, which is 4.
23. R is a P, but I does not necessarily have a P in it (I is a P, but you cannot assume it has a P).
24. X does not overriding the original method in W, and so p will call the parents x and not the childs.
25. Y overrides the original method in W, and so p will call the childs x and not the parents, however `super.p` will call the original p.
26. the sum = $37+5+7+7+5=61$.
27. A priority queue is held as a min heap binary balanced tree, so every child is greater than its parent. It doesn't print out sorted, but rather where each parent is greater than both children and based on the order it is entered.
28. $m(abcde) = cde + m(cde) = cde + e + m(e) = cde + e + e = cdeee$
29. $m(sturges) = turges + m(turges) = turges + urges + m(urges) = turges + urges + ges + m(ges) = turges + urges + ges + ges = turgesurgesgesgesges$

30. The equation is the same as saying $145|x$.
31. The pre-order traversal will print the node, then go left, then go right.
32. Every time a value is less than peek, the stack will pop off the values greater than the values and put them into a ArrayList that is behaving like a queue. This means that whatever is getting popped off is getting reversed when it is put back on. That is why it is not quite sorted.
33. I would prefer to use "\|\|" but when I tried to write the distractor "[]" I found it worked as well, so I went with that.
34. the keyset is TreeSet that will only list the key values in alpha order.
35. In a directional graph, the number of edges is the number of unique arrows pointing at another node. If you look at the code, each edge is described, 13 lines of code = 13 edges.
36. A simple path in a graph is a path that can trace from one point in the graph to another along its edges without repeat. If there are repeats, then it is just a path.
37. here is the truth table:

A	B	C	$B \mid C$	$! (B \mid C)$	$A \& \& ! (B \mid C)$
0	0	0	0	1	0
0	1	0	1	0	0
1	0	0	0	1	1
1	1	0	1	0	0
0	0	1	1	0	0
0	1	1	1	0	0
1	0	1	1	0	0
1	1	1	1	0	0

38. $E - A * (A + F - B) * (D + B) / (C - B)$
 $E - A * (- + AFB) * (+DB) / (-CB)$
 $E - (/ ** A - + AFB + DB - CB)$
 $- E / ** A - + AFB + DB - CB$
39. $ABC + AB \Rightarrow AB(C+1) \Rightarrow AB(1) \Rightarrow AB$
40. To find -56 as a byte, you must use two's compliment.
56 $\rightarrow 00111000$
flip $\rightarrow 11000111$
add 1 $\rightarrow 11001000 \Rightarrow -56$