Name_____ School_____ 1A 2A 3A 4A 5A 6A

COMPUTER SCIENCE WRITTEN TEST

FEBRUARY 18, 2017

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.

COMPUTER SCIENCE WRITTEN TEST – FEBRUARY 18, 2017

Note: Correct responses are based on **Java SE Development Kit 8 (JDK 8)** from Sun Microsystems, 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.

Which of the following is equivalent to $16_8 + 32_{16}$?

A) 110000_2 **B)** 32_8 **C)** 64_{10} **D)** 64_{16} **E)** 48_{24}

Question 2.

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

A) 9.0 **B)** 9.1 **C)** 9.9 **D)** 10.0

E) No output due to an error.

```
double x = 0.1;
int y = (int) 9.9;
double z = (int) (x + y);
out.println(z);
```

Question 3.

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

A) p"r"+ **B)** +"p"'' **C)** +rp"+

D) +r+p+ **E)** No output due to an error.

```
String p = "+";
char r = '"';
out.println(p + "r" + 'p' + r + '+');
```

Question 4.

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

A) cbcabcabc **B)** abcabcabc **C)** acacac

D) acabcabc **E)** cbbcbcbbcbcbbcb

```
String s = "abcabcabc";
s.replaceAll("a", "c");
s.replaceAll("cb", "a");
out.println(s);
```

Question 5.

Which of the following conditions will cause the Boolean expression to the right to evaluate to FALSE?

A) P = FALSE; Q = FALSE; R = TRUE

B) P = FALSE; Q = TRUE; R = TRUE

C) P = TRUE; Q = FALSE; R = TRUE

D) P = TRUE; Q = TRUE; R = FALSE

E) P = TRUE; Q = TRUE; R = TRUE

$$P + \overline{Q} * R + \overline{Q}$$

Question 6.

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

A) 0.0 **B)** 0.75 **C)** 1.0 **D)** 2.0

E) No output due to an error.

```
int alfa = 3;
int bravo = 4;
out.println(Math.ceil(alfa / bravo));
```

Question 7.

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

A) 127

B) 4540forty-two

C) 85forty-two

D) 88forty-two

E) No output due to an error.

```
int eger = 42;
float ingPoint = 40.2f;
short answer = (short) ingPoint;
short erAnswer = answer;
erAnswer /= 10 + 2;
eger += erAnswer;
String ofText = "forty-two";
out.println(eger + answer + ofText);
```

Question 8.

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

A) WXZ **B)** WYZ **C)** XY

D) WX **E)** WY

```
int year = 2017;
if (year % 4 != 0) { out.print("W"); }
if (year % 4 == 0) { out.print("X"); }
else if (year % 4 != 0) { out.print("Y"); }
else { out.print("Z"); }
```

Question 9.

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

A) `science about telescopes`

B) `science is about about`

C) `science no about than is telescopes`

D) `science is more astronomy`

E) `Computer science nomore astronomyis`

```
String dijkstra = "Computer science is no"
    + "more about computers than astronomy"
    + "is about telescopes.";
String[] quote = dijkstra.split(" ");
for (int i = 1; i < quote.length; i *= 2) {
  out.print(quote[i] + " ");
}
```

Question 10.

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

A) `Dustin` **B)** `Eleven` **C)** `Lucas`

D) `Mike` **E)** `Will`

```
String[] things = { "Dustin", "Eleven",
                    "Lucas", "Mike" };
int x = Arrays.binarySearch(things, "Will");
int strange = ~x;
int stranger = strange + ~0;
out.println(things[stranger]);
```

Question 11.

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

A) `best times worst times`

B) `best times, worst times,`

C) `of it of ...`

D) `of it of`

E) No output due to an error.

```
String tale = "It was the best of times, ";
tale += "it was the worst of times, ...";

Scanner reader = new Scanner(tale);
while (reader.hasNext()) {
  if (reader.next().length() > 3) {
    out.print(reader.next() + " ");
  }
}
```

Question 12.

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

A) `3 9 27`

B) `3 9 27 81`

C) `1 3 9 27 81 243`

D) `1 3 9 27`

E) `1 3 9 27 81`

```
boolean go = true;
int tri = 1;
while (go) {
  out.print(tri + " ");
  tri *= 3;
  if (tri / 2 < 100) { go = !go; }
  if (tri * 2 < 150) { go = !go; }
}
```

Question 13.

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

A) `A` **B)** `B` **C)** `C` **D)** `D` **E)** `E`

```
boolean one = 5 + 4 / 3 * 2 > 5 * 4 / 3 + 2;
boolean two = one || !one && one;
if (!two) {
  if (one) { out.print("A"); }
  else { out.print("B"); }
}
else if (one || two) {
  if (!two) { out.print("C"); }
  else { out.print("D"); }
}
else { out.print("E"); }
```

Question 14.

Which of the following data types could correctly replace **<#1>** in the code segment to the right?

A) `char[]` **B)** `String` **C)** `char`

D) `Double` **E)** `int`

```
char digit6 = '6';
char digit8 = '8';
char digit9 = '9';

<#1> temp = digit9 + digit8 + '.' + digit6;
```

Question 15.

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

A) `0` **B)** `1` **C)** `2` **D)** `3`

E) No output due to an error.

```
List<String> cat = new ArrayList<>();
cat.add("cat");
List<String> dog = new LinkedList<>(cat);
dog.add("dog");
List<String> bird = new ArrayList<>(dog);
bird.add("bird");
out.println(bird.size());
```

Question 16.

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

A) 84 **B)** 127 **C)** 128 **D)** 190 **E)** 254

```
int base = 127;
base = base >> 6 << 8 >> 2 << 1;
out.println(base);
```

Question 17.

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

A) XZ9

B) 0YZ9

C) XYZ9

D) YZ

E) No output due to an error.

```
int[] d = { 3, 13, 11, 9, 12, 5 };
try {
  int a = d[(d[0] + d[d.length - 1]) / 2];
  int b = d[0] + d[d[0]] - a;
  out.print(a / b);
  out.print(d[d[d[0]]]);
} catch (RuntimeException re) {
  out.print("X");
} catch (Exception e) {
  out.print("Y");
} finally {
  out.print("Z");
}
out.println(d[d[0]]);
```

Question 18.

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

A) 12

B) 23 31

C) 12 31

D) 31

E) 12 23 31

```
int oneTwo = "one".compareTo("two");
int twoThree = "two".compareTo("three");
int threeOne = "three".compareTo("one");
if (oneTwo < 0) { out.print("12 "); }
if (twoThree < 0) { out.print("23 "); }
if (threeOne < 0) { out.print("31"); }
```

Question 19.

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

A) ABCD

B) ABCDCBA

C) AABAABCBAABCDCBA

D) DCBA

E) ABACABADABACABA

```
Stack<Character> left = new Stack<>();
Stack<Character> right = new Stack<>();
String text = "ABCD";
for (int i = 0; i < text.length(); i++) {
  left.push(text.charAt(i));
  while (!right.isEmpty()) {
    left.push(right.pop());
  }
  right.addAll(left);
}
while (!left.isEmpty()) {
  out.print(left.pop());
}
```

Question 20.

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

A) J u **B)** u J **C)** n J

D) a V **E)** V a

```
String s = "JohnVonNeumann";
char[] c = s.toCharArray();
char max = Character.MIN_VALUE;
char min = Character.MAX_VALUE;
for (int i = 0; i < c.length; i++) {
  max = (char) Math.max(max, c[i]);
  min = (char) Math.min(c[i], min);
}
out.println((char) max + " " + (char) min);
```

Question 21.

Which of the following strings is fully matched by the regular expression to the right?

A) 1/a/ /b/2

B) 3/x//y/4

C) 5/q/_/p/d

D) d/0/ /_/2

E) d/\w/\s/\w/\d

```
d/\w/\s/\w/\d
```

Question 22.

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

A) 27862596 **B)** 27,862,596 **C)** 29,868,702

D) 29868702 **E)** No output due to an error.

```
String population = "27,862,596";
double rateOfGrowth = 1.072;
double projected = rateOfGrowth;
projected *= Integer.parseInt(population);
out.print((int) projected);
```

Question 23.

What is the output of line **<#1>** in the **Client Code** to the right?

A) –1 **B)** 1 **C)** 3 **D)** 5 **E)** 7

Question 24.

What is the output of line **<#2>** in the **Client Code** to the right?

A) `[9, 9, 8, 5, 4, 2, 0]`

B) `[9, 8, 5, 4, 2, 0, 9]`

C) `[4, 9, 8, 2, 5, 0, 9]`

D) `[0, 2, 4, 5, 8, 9]`

E) `[0, 2, 4, 5, 8, 9, 9]`

Question 25.

Which of the following algorithms is implemented by the `alpha()` method to the right?

A) Binary Search **B)** Sequential Search

C) Selection Sort **D)** Insertion Sort

E) Swap

Question 26.

What is the expected runtime performance for the `alpha()` method in the worst case? Choose the most restrictive answer.

A) O(1) **B)** $O(\log_2 N)$ **C)** O(N)

D) $O(N * \log_2 N)$ **E)** $O(N^2)$

Question 27.

Which of the following algorithms is implemented by the `beta()` method to the right?

A) Binary Search **B)** Sequential Search

C) Selection Sort **D)** Insertion Sort

E) Swap

Question 28.

What is the expected runtime performance for the `beta()` method in the worst case? Choose the most restrictive answer.

A) O(1) **B)** $O(\log_2 N)$ **C)** O(N)

D) $O(N * \log_2 N)$ **E)** $O(N^2)$

Question 29.

Which of the following algorithms is implemented by the `gamma()` method to the right?

A) Binary Search **B)** Sequential Search

C) Selection Sort **D)** Insertion Sort

E) Swap

Question 30.

What is the expected runtime performance for the `gamma()` method in the worst case? Choose the most restrictive answer.

A) O(1) **B)** $O(\log_2 N)$ **C)** O(N)

D) $O(N * \log_2 N)$ **E)** $O(N^2)$

```
static int alpha(int[] a, int b, int c) {
  int d = 0;
  for (int e = b; e < c; e++) {
    int f = beta(a, e, c);
    if (e != f) { d += gamma(a, e, f); }
  }
  return d;
}

static int beta(int[] a, int b, int c) {
  int d = b;
  for (int e = b; e < c; e++) {
    if (a[e] > a[d]) { d = e; }
  }
  return d;
}

static int gamma(int[] a, int b, int c) {
  int d = a[b];
  int e = a[c];
  a[b] = e;
  a[c] = d;
  return 1;
}
```

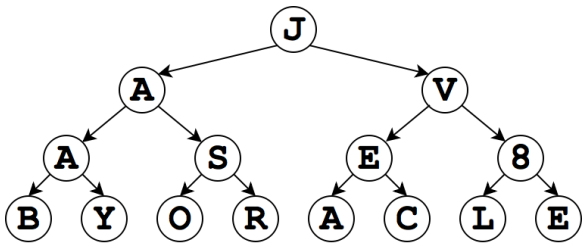
Client Code

```
int[] x = { 4, 9, 8, 2, 5, 0, 9 };
int y = 0;
int z = x.length;

out.println(alpha(x, y, z));     //<#1>
out.print(Arrays.toString(x));   //<#2>
```

Question 31.

Which of the following is a pre-order traversal of the binary tree to the right?



A) JAVASE8BYORACLE

B) JAABYSORVEAC8LE

C) BAYAOSRJAECVL8E

D) 8AAABCEEJLORSVY

E) BYAORSAACELE8VJ

Question 32.

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

A) 20 **B)** 27

C) 34 **D)** 82

E) No output due to an error.

```
int[][] m = { { 7, 7, 9, 8, 3 },
              { 1, 4, 2, 6, 9 },
              { 3, 6, 1, 4, 8 } };
int n = 0;
for (int r = 0; r < m.length - 1; r++) {
  for (int c = 1; c < m[0].length; c++) {
    int s = m[r][c - 1] + m[r + 1][c - 1] +
            m[r][c]     + m[r + 1][c];
    if (s > n) { n = s; }
  }
}
out.println(n);
```

Question 33.

What is the output of line **<#1>** in the code segment to the right?

A) K **B)** i **C)** r

D) [K, i] **E)** null

Question 34.

What is the output of line **<#2>** in the code segment to the right?

A) 2 **B)** 12 **C)** 14

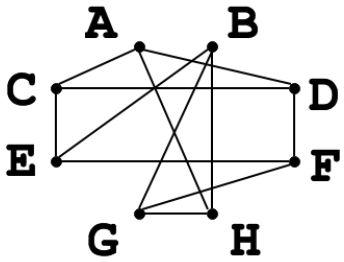
D) 16 **E)** 24

```
Map<Character, Character> map;
map = new TreeMap<>();
String jack = "Jack_Sparrow";
String kidd = "Captain_Kidd";
for (int i = 0; i < kidd.length(); i++) {
  map.put(jack.charAt(i), kidd.charAt(i));
  map.put(kidd.charAt(i), jack.charAt(i));
}
out.println(map.get(map.get('r')));    //<#1>
out.println(map.size());               //<#2>
```

Question 35.

What is the edge connectivity of the graph to the right?

A) 1 **B)** 3 **C)** 4 **D)** 8 **E)** 12



Question 36.

What is the vertex connectivity of the graph to the right?

A) 1 **B)** 3 **C)** 4 **D)** 8 **E)** 12

Question 37.

Which of the following digital components does the logic gate diagram to the right represent?

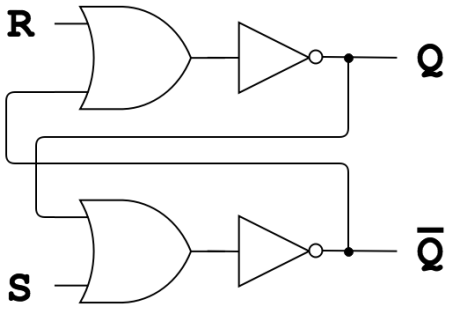
A) Decoder

B) Back Propagator

C) Flip-flop

D) Multiplexer

E) The component is invalid.



Question 38.

Which of the following Boolean expressions simplifies to a value of Q?

A) P * (Q + $\overline{Q}$) **B)** P * (Q + $\overline{P}$) **C)** P + $\overline{P}$ * Q **D)** P + $\overline{Q}$ * P **E)** Q * (P + $\overline{P}$)

FREE RESPONSE QUESTIONS

Question 39.

What is the postfix notation (i.e., reverse Polish notation) of the arithmetic expression to the right?

Write your answer on the answer sheet.

```
(A - (B + C) * (D - E))
```

Question 40.

What is the 8-bit, signed, 2's complement binary representation of `n`, as shown in the code segment to the right?

Write your answer on the answer sheet.

```
byte n = -36;
```

★ DOUBLE-CHECK YOUR ANSWERS ★

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)
```

