A+ Computer Science
M/C Written Test

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 12 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.

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, 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.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.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)

A+ Computer Science Contest #2425-05 November 02, 2024

Note: Correct responses are based on **Java SE Development Kit 20 (JDK 20)** 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 20 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 19_{10} - 21_4 ?

A. 1010_2 B. 1100_2 C. 12_{10} D. 13_8 E. C_{16}

QUESTION 2

What is output by the code to the right?

A. 42 B. 58 C. 28 D. 36 E. 23.6

```
out.print(4 + 8 * 5 - 2);
```

QUESTION 3

What is output by the code to the right?

A. I have 2.466 gallons of milk.
B. I have 2.4 gallons of milk.
C. I have 2.5 gallons of milk.
D. I have 2.466 gallons %s
E. I have %.1f gallons %s

```
String s = "I have %.1f gallons %s";
out.printf(s, 2.466, "of milk.");
```

QUESTION 4

What is output by the code to the right?

A. Billiard Ball
B. Williamiard Ball
C. Williamiard Bill
D. Williamiard William
E. repiard Ball

```
String s = "Billiard Ball";
String rep = "William";
out.print(s.replace("Bill", rep));
```

QUESTION 5

What is output by the code to the right?

A. true B. false

```
boolean a = true;
out.print(a||false&&(true||false));
```

QUESTION 6

What is output by the code to the right?

A. 5 B. 4
C. 1
D. one
E. four minus five

```
out.print(Math.subtractExact(5, 4));
```

QUESTION 7

What is output by the code to the right?

A. 3.5hello97 B. 3.5ahello
C. 3.5helloa
D. Output cannot be determined.
E. There is no output due to a compile error.

```
int a = 5;
double b = 1.5;
char c = 'a';
String d = "hello";
out.print(a - b + d + c);
```

QUESTION 8

What is output by the code to the right?

A. > B. = C. <= D. >=

E. ><=

```
int j = 48; char b = '0';
if(j>=b) out.print(">");
else if(j<=b) out.print("<");
if(j==b) out.print("=");
```

QUESTION 9

How many asterisks are output by the code to the right?

A. 5 B. 4 C. 15 D. 10 E. 0

```
for(int i = 0; i < 5; i++) {
  for(int j = 0; j <= i; j++) {
     out.print("*");
  }
  out.println();
}
```

QUESTION 10

What is the value of `b` after the code to the right is executed?

A. `[0, 0, 0, 0]` B. `[0, 3, 0, 0]`

C. `[0, 0, 3, 0]` D. `[3, 3, 3, 3]`

E. This cannot be determined until runtime.

```
int [] a = new int[4];
int [] b = {0, 1, 2, 3};
b = a;
a[2] = 3;
```

QUESTION 11

What is the output by the code to the right?

A. `Hello`

B. `Hello World!`

C. `HelloWorld`

D. `HelloWorld!`

E. There is no output due to a runtime exception.

```
String s = "Hello World!";
Scanner file = new Scanner(s);
out.print(file.next()+file.next());
```

QUESTION 12

What is the output by the code to the right?

A. 24 B. 48 C. 120 D. 240

E. There is no output due to a runtime exception.

```
int i = 2;
for(int k = 1; k < 5; k++) {
  i*=k;
}
out.print(i);
```

QUESTION 13

What is order of precedence for the operations on the right from highest precedence to lowest precedence?

A. I, II, III

B. I, III, II

C. II, III, I

D. III, I, II

E. III, II, I

I. =(assignment)
II. +(additive)
III. &&(logical)

QUESTION 14

What is the output by the code to the right?

A. 8 B. 64 C. 32 D. 16 E. 4

```
out.print(Float.BYTES);
```

QUESTION 15

What is the output by the code to the right?

A. `hi`

B. `45.24`

C. `1`

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

E. There is no output due to a runtime exception.

```
ArrayList <Object> ar;
ar = new ArrayList();
ar.add("hi");
ar.add(45.24);
ar.add(1);
out.print(ar.get(1));
```

QUESTION 16

What is the output by the code to the right?

A. `13` B. `15` C. `11` D. `17` E. `435`

```
out.print(8|7&29);
```

QUESTION 17

What is the output by the code to the right?

A. `true` B. `false`

```
boolean a = true;
boolean b = !a;
a = false&&b||(true&&false);
out.print(!(b||a));
```

QUESTION 18

What is the output by the code to the right?

A. `0 [2, 5]`

B. `2 [5, 0]`

C. `true [2, 5]`

D. `true [5, 0]`

E. There is no output due to a runtime exception.

```
ArrayList <Integer> b;
b = new ArrayList();
b.add(2); b.add(5); b.add(0);
out.print(b.remove(0) + " " + b);
```

QUESTION 19

What is the output by the code to the right?

A. `1998927` B. `1010148` C. `1909037`

D. There is no output due to an infinite loop.

E. There is no output due to a runtime exception.

```
int a = 7298991;
while(a > 1) {
  out.print(a%10);
  a /= 10;
  a += 1;
}
```

QUESTION 20

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

A. `7` B. `8` C. `9` D. `10` E. `256`

```
int i = 2;
do {
  out.print('*');
  i*=2;
} while(i < 257);
```

QUESTION 21

What is the output of the function call `ret(3, 3)`?

A. `hi` B. `hello`

C. `evening` D. `hihello`

E. `hihelloevening`

QUESTION 22

What is the output of the function call `ret(39, 6)`?

A. `hi` B. `hello`

C. `evening` D. `hihello`

E. `hihelloevening`

```
// Use the code below to answer
// questions 21-22.

public static String ret(int a, int
b) {
  if(a + b == 45) return "hi";
  if(a + b < 45) return "hello";
  if(a + b > 45) return "evening";
  else return "this shouldn't be a
possible output";
}
```

QUESTION 23

What should replace **<code 1>** in the code to the right in order for `Geode` to be a subclass of `Rock`?

A. `defines`

B. `extends`

C. `implements`

D. `inherits`

E. Nothing, this space can be left blank.

QUESTION 24

What should replace **<code 2>** in the code to the right in order for `Geode` constructor to function properly?

A. `super(w, n);`

B. `super(3.2, n);`

C. **<code 2>** can be left blank.

D. Only A and B.

E. All of A, B, and C.

```
// Use the code below to answer
// questions 23-26.

class Rock{
  double weight;
  String name;
  public Rock(double w, String n) {
     weight = w;
     name = n;
  }
  String getName() {
     return name;
  }
}

class Geode <code 1> Rock{
  String formation;
  public Geode(double w, String n,
String f) {
     <code 2>
     formation = f;
  }
  String getName() {
     return super.getName() + " " +
formation;
  }
}
```

QUESTION 25

Assuming the blanks above have been properly filled, what is the output of the code marked **`//line 1`**?

A. `chert`

B. `2.34`

C. `name`

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

E. There is no output due to a runtime exception.

QUESTION 26

Assuming the blanks above have been properly filled and any errors fixed, what is the output of the code marked **`//line 2`**?

A. `Snowball`

B. `Chalcedony Snowball`

C. `Snowball Chalcedony`

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

E. There is no output due to a runtime exception.

```
// use the code above and the code
// below to answer questions 25-26

// client code
//////////////CLIENT CODE/////////////
a = new Rock(2.34, "chert");
Rock b = new Geode(4.23, "Snowball",
"Chalcedony");
out.println(a.getName()); //line 1
out.println(b.getName()); //line 2
```

QUESTION 27

What is the result of the function call `rec(3,3)`?

A. `7` B. `9` C. `6` D. `10`

E. There is no output due to a runtime exception.

QUESTION 28

What is the result of the function call `rec(5,5)`?

A. `15`

B. `21`

C. `19`

D. `25`

E. There is no output due to a runtime exception.

```
public static int rec(int a, int b) {
  if(a == 1 || b == 1) return
Math.max(a, b);
  if(Math.pow(a, b) > Math.pow(b, a))
{
     return rec(a - 1, b) + b - 1;
  }
  else {
     return rec(a, b - 1) + a - 1;
  }
}
```

QUESTION 29

What is output by the code to the right?

A. `123` B. `173` C. `83` D. `131`

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

```
out.println(Integer.toString(123,8));
```

QUESTION 30

What is output by the code to the right?

A. `hi`

B. `hey`

C. `hello`

D. `good evening`

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

```
Map<String, String> map = new
TreeMap<String, String>();
map.put("hi", "hello");
map.put("hey", "hello");
map.put("hello", "good evening");
map.put("good evening", "hi");
out.print(map.get("hello"));
```

QUESTION 31

What is output by the code to the right?

A. `[2, 1, 0]`

B. `[3, 2, 0]`

C. `[4, 0, 0]`

D. `[0, 1, 2]`

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

```
int [][] a = new int [3][3];
for(int i = 0; i < 3;i++) {
  for(int j = 0; j <= i; j++) {
     a[j][2-i] = i + j;
  }
}
out.print(Arrays.toString(a[2]));
```

QUESTION 32

What is output by the code to the right?

A. `4`

B. `3`

C. `1`

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

E. There is no output due to a runtime exception.

```
Stack<Integer> s = new Stack();
s.add(4); s.add(3); s.add(1);
s.peek();
out.print(s.peek());
```

QUESTION 33

What is output by the code to the right?

A. `hi`

B. `hello`

C. `hey`

D. `hola`

E. `howdy`

```
List<String> l = new LinkedList();
l.add("hi");
l.add("hello");
l.add("hey");
l.add(2, "hola");
l.add("howdy");
l.remove(3);
l.remove("hi");
out.print(l.get(2));
```

QUESTION 34

What is output by the code to the right?

A. `noodles`

B. `bob`

C. `nbobdlbobs`

D. `nbobbobdlbobs`

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

```
String s = "noodles";
out.print
    (s.replaceAll("[aeiou]+", "bob"));
```

QUESTION 35

What is output by the code to the right?

A. `0` B. `56` C. `42` D. `60`

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

```
out.print(4&6|7<<3);
```

QUESTION 36

What is the average time complexity for insertion on an array of size `n`?

A. `O(1)` B.`O(n)` C.`O(log(n))` D. `O(nlog(n))` E. `O(n^2)`

QUESTION 37

Which of the following Java boolean expressions is equivalent to the truth table on the right?

A. `A || !B`

B. `B || !A`

C. `A && !B`

D. `B && !A`

E. `A || B && A`

A	B	Result
True	True	False
True	False	True
False	True	False
False	False	False

QUESTION 38

What is the minimum number of edges in a minimum spanning tree of a graph with 5 vertices?

A. `3`

B. `4`

C. `5`

D. `6`

E. `7`

QUESTION 39

What is the value of the postfix expression to the right? Assume integer division.

```
8 3 + 2 * 2 1 + / 7 * 4 /
```

QUESTION 40

What is the best case (minimum) height of a binary search tree with 14 nodes?