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 23 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 #2526-12 February 14, 2026

Note: Correct responses are based on **Java JDK 23** 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 JDK 23 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 the expression 33_6?

A. 26_8 B. 110_4 C. 22_{10} D. 15_{16} E. 211_3

QUESTION 2

What is output by the code to the right?

A. -9 B. 2

C. -6 D. -67

E. There is no output due to an error.

```
out.println(30 / 6 - 14);
```

QUESTION 3

What is output by the code to the right(&'s indicate blank spaces)?

A. &&&5

B. &&53

C. 5&&&

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

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

```
char c = '5';
out.printf("%4d",c);
```

QUESTION 4

What is output by the code to the right?

A. sideDod
B. sideDowd
C. psideDs
D. psideDos
E. There is no output due to an error.

```
String s = "TheUpsideDown";
s = s.substring(5, 11);
s += s.charAt(2);
out.println(s);
```

QUESTION 5

What is output by the code to the right?

A. true B. false

```
boolean a = true;
boolean b = true;
b = a | b ^ a;
out.println(b);
```

QUESTION 6

What is output by the code to the right?

A. 6.0 B. 6.5 C. 7.0 D. 7

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

```
int c = 7;
double d = 6.5;
d = Math.max(c, d);
out.println(d);
```

QUESTION 7

What is the output by the code to the right?

A. 23 B. 25 C. 45

D. 15 E. 235

```
int a = 9;
if(a < 10)
  out.print(2);
else if(a > 8)
  out.print(3);
else
  out.print(4);
out.print(5);
```

QUESTION 8

What is the output by the code to the right?

A. `7 22 22`
B. `7 29 29`
C. `9 33 37`
D. `8 31 37`
E. There is no output due to an infinite loop.

```
int i = 0, j = 15, k = 1;
while(k < j) {
  j += 2;
  k += i++;
}
out.println(i+" "+j+" "+k);
```

QUESTION 9

What is output by the code to the right?

A. `76`
B. `47`
C. `52`
D. `53`
E. There is no output due to an error.

```
int a = 1;
for(int y = 4; y < 100; y *= 3)
  a = y + a;
out.println(a);
```

QUESTION 10

What is the output by the code to the right ?

A. `46`
B. `67`
C. `0`
D. `42`
E. There is no output due to an error.

```
int[] i = new int[] {
        34, 54, 29, 1, 8, 13, 17,
        23, 56, 84, 98, -120, 35};
int b = i[2]+i[6];
out.println(b);
```

QUESTION 11

What is the output by the code to the right?

A. `16`
B. `11`
C. `18`
D. `10`
E. There is no output due to an error.

```
String r = "srrsrsrrrrsssr";
r += "rrssssrsrrsrsrssrssrs";
Scanner sc = new Scanner(r);
sc.useDelimiter("r");
int num = 0;
while(sc.hasNext()) {
  num++;
  sc.next();
}
out.println(num);
```

QUESTION 12

What is the output by the code to the right?

A. `60`
B. `63`
C. `55`
D. `51`
E. There is no output due to an error.

```
int sum = 0;
for(int i = 0; i < 15; i+=3)
  sum += i * 2;
out.println(sum);
```

QUESTION 13

What is the order of precedence for the operators to the right ?

A. `II, IV, III, I`
B. `II, III, I, IV`
C. `III, II, IV, I`
D. `III, II, I, IV`
E. `II, III, IV, I`

```
I. <=
II. ~
III. /
IV. >>>
```

QUESTION 14

What is the output by the code to the right ?

A. `832`
B. `8`
C. `1632`
D. `64`
E. `3232`

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

QUESTION 15

What is the output by the line marked `//q15` in the code to the right?

A. `[212, 3, -8]` B. `[212, 3, -8, 13]`

C. `[3, -8, 13]` D. `[3, -8, 13, 67]`

E. There is no output due to an error.

QUESTION 16

What is the output by the code to the right?

A. `2` B. `5`

C. `6` D. `8`

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

```
ArrayList<Integer> a;
a = new ArrayList<Integer>();
a.add(9);a.add(17);
a.add(212);a.add(3);
a.add(-8);a.add(13);
a.add(67);a.add(-40);
a.add(-12);a.add(5);
out.println(a.subList(2, 5)); //q15
a.removeIf(n -> n % 3 != 1);
out.println(a.size()); //q16
```

QUESTION 17

What is the output by the code to the right?

A. `16` B. `25`

C. `21` D. `31`

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

```
int a = 20;
int b = 15;
int c = ~a + b;
out.println(~c + a);
```

QUESTION 18

What is the output by the line marked `//q18` code to the right?

A. `620` B. `610`

C. `510` D. `1010`

E. There is no output due to an error.

QUESTION 19

What is the output by the line marked `//q19` code to the right?

A. `622` B. `428`

C. `442` D. `342`

E. There is no output due to an error.

```
int n = 0b100101101;
String s = Integer.toString(n, 7);
n = Integer.parseInt(s, 8);
s = Integer.toOctalString(n);
out.println(s); //q18
n = Integer.parseInt(s);
s = Integer.toHexString(n);
n = Integer.parseInt(s, 12);
s = Integer.toString(n, 9);
n = Integer.parseInt(s);
out.println(n); //q19
```

QUESTION 20

What is the output by the code to the right?

A. `8888` B. `XX`

C. `88X` D. `X88`

E. There is no output due to an error.

```
char x = 'X';
int i = 0;
out.print(true  ? x : 0);
out.print(false ? i : x);
```

QUESTION 21

What is the output by the code to the right?

A. `8`

B. `11`

C. `4`

D. `6`

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

```
String reg = "\\w\\W?[^f-m3-7]{3,6}";
String s = "Aei883 2rf 2f09 23f-09j";
s+= "*iDIi73 ^2Vid3#iun23 O@#noin32";
Pattern p = Pattern.compile(reg);
Matcher m = p.matcher(s);
count = 0;
while(m.find())
  count++;
out.println(count);
```

QUESTION 22

Which of the following could replace **<1*>** in the code to the right so that the `A` constructor works as intended?

A. `self.`
B. `this.`
C. `A.`
D. Nothing is required.
E. More than one of the above.

QUESTION 23

Which of the following terms is demonstrated by the `add` methods in the `A` and `B` classes to the right?

A. Overriding
B. Overloading
C. Encapsulation
D. A and B.
E. All of the above.

QUESTION 24

Assuming **<1*>** has been filled correctly, what is output by the lines marked `//q24` in the client code to the right?

A. `a: 4 b: 5 c: 6`
B. `a: 4 b: 9 c: 11`
C. `a: 4 b: 9 c: 6`
D. There is no output due to a compile error.
E. There is no output due to a runtime error.

QUESTION 25

Assuming **<1*>** has been filled correctly, what is output by the lines marked `//q25` in the client code to the right?

A. `a: 4 b: 9 c: 16`
B. `a: 4 b: 13 c: 16`
C. `a: 4 b: 14 c: 17`
D. There is no output due to a compile error.
E. There is no output due to a runtime error.

QUESTION 26

Assuming **<1*>** has been filled correctly, what is output by the lines marked `//q26` in the client code to the right?

A. `a: 4 b: 36 c: 80`
B. `a: 4 b: 52 c: 80`
C. `a: 4 b: 56 c: 85`
D. There is no output due to a compile error.
E. There is no output due to a runtime error.

```
class A{
 int n;
 String s;
 public A(String s, int n) {
  <1*>n = n;
  <1*>s = s;
 }
 public int add() {
  return n++;
 }
 public String toString() {
  return s+": "+n;
 }
}
class B extends A{
 public B(String s, int n) {
  super(s, n * 2);
 }
 public int add(int i) {
  n = add() + i;
  return n;
 }
 public int mult(int i) {
  n *= i;
  return i;
 }
}
//////////////client code///////////
A a = new A("a", 3);
B b = new B("b", 4);
A c = new B("c", 5);
a.add();
b.add();
c.add();
String s = a + " ";
s += b + " " + c;
out.println(s); //q24
b.add(4);
c.add(5);
s = a + " ";
s += b + " " + c;
out.println(s); //q25
b.mult(4);
c.mult(5);
s = a + " ";
s += b + " " + c;
out.println(s); //q26
```

QUESTION 27

What is output by the code to the right?

A. `Helloworld`

B. `Hell`
`o world`

C. `Hell`
`O World`

D. `Hello world`

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

```
out.print("Hell");
out.println("o world");
```

QUESTION 28

What could replace **<?*>** in the code to the right so that the code compiles without error?

A. `add`

B. `offer`

C. `push`

D. A and B.

E. All of the above.

QUESTION 29

What is output by the client code to the right?

A. `[Manatee, Zebra, Ocelot, Leopard]`

B. `[Manatee, Ocelot, Tortoise, Zebra]`

C. `[Manatee, Ocelot, Zebra, Tortoise]`

D. `[Lemur, Anaconda, Tortoise, Elephant]`

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

```
PriorityQueue<String> p;
p = new PriorityQueue<String>();
p.<?*>("Lemur");
p.<?*>("Anaconda");
p.<?*>("Tortoise");
p.<?*>("Elephant");
p.<?*>("Giraffe");
p.<?*>("Hippo");
p.<?*>("Manatee");
p.<?*>("Zebra");
p.<?*>("Ocelot");
p.<?*>("Leopard");
p.poll();
p.poll();
p.poll();
p.poll();
p.poll();
p.poll();
out.println(p);
```

QUESTION 30

What is output by the line marked `//q30` in the client code to the right?

A. `EEEEECABBDD`

B. `EECCABBDDD`

C. `ACBBCEDDED`

D. `DEDDECBBCA`

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

QUESTION 31

What is output by the line marked `//q31` in the client code to the right?

A. `EDDECDDECDDECDDECBBCA`

B. `EECECECECCABBDDDDDDDD`

C. `ACBBCEDDCEDDCEDDCEDDE`

D. `EEECECECCABBDDDDDDDD`

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

```
public String recur(int v) {
 if(v <= 0)
  return "A";
 if(v % 4 == 1)
  return "C" + recur(v - 3);
 if(v % 3 == 2)
  return "E" + recur(v - 5);
 if(v < 10)
  return recur(v / 2) + "B";
 if(v > 30)
  return recur(v - 20) + "D";
 return "E" + recur(v - 7);
}
//////////////client code/////////////
out.println(recur(79)); //q30
out.println(recur(203)); //q31
```

QUESTION 32

Which of the following could replace **<1*>** in the code to the right so that the code can compile and execute without error and as intended?

A. `Object`
B. `T`
C. `E`
D. A and C only
E. A, B, and C

QUESTION 33

What is output by the line marked `//q33` code to the right?

A. `Fig`
B. `Dragonfruit`
C. `Carrot`
D. `Banana`
E. There is no output due to a runtime error.

QUESTION 34

What is output by the line marked `//q34` code to the right?

A. `[Dragonfruit, Banana, Apple]`
B. `[Fig, Eggplant, Dragonfruit]`
C. `[Eggplant, Banana, Carrot]`
D. `[Banana, Carrot, Eggplant]`
E. There is no output due to a runtime error.

QUESTION 35

What is output by the line marked `//q35` code to the right?

A. `[15, 14, 20, 12, 11, 14, 2]`
B. `[15, 26, 14, 13, 20, 12, 11]`
C. `[26, 14, 20, 12, 11, 14, 2]`
D. `[26, 14, 13, 20, 12, 11, 14]`
E. There is no output due to a runtime error.

QUESTION 36

Which of the following data structures is implemented by the `Structure` class to the right ?

A. `Map`
B. `Priority Queue`
C. `Stack`
D. `Queue`
E. `Linked List`

```
class Structure< <1*> >{

 ArrayList< <1*> > list;
 int size;

 public Structure() {
  list =new ArrayList< <1*> >();
 }

 public void add(<1*> e) {
  size++;
  list.add(0, e);
 }

 public <1*> remove(int i) {
  size--;
  return list.remove(i);
 }

 public <1*> remove() {
  size--;
  return list.remove(0);
 }

 public String toString() {
  return list.toString();
 }
}
//////////////client code/////////////
Structure<String> s1;
Structure<Integer> s2;
s1 = new Structure<String>();
s2 = new Structure<Integer>();
s1.add("Apple");
s1.add("Banana");
s1.add("Carrot");
s1.remove();
s1.add("Dragonfruit");
s1.add("Eggplant");
s1.remove();
s1.add("Fig");
String s = s1.remove();
out.println(s); //q33
out.println(s1); //q34
s2.add(17);
for(int y = 0; y < 15; y++) {
 if(y % 3 == 1)
  s2.add(y * 2);
 if(y / 4 == 2)
  s2.add(y + 3);
 if(y % 5 == 0)
  s2.remove();
 if(y / 3 > 4)
  s2.add(y);
}
out.println(s2); //q35
```

QUESTION 37

What is the sum of all the popped items in the stack to the right?

A. `91`
B. `222`
C. `263`
D. `50`
E. `-8`

QUESTION 38

What is the sum of all the unpopped items in the stack to the right?

A. `91`
B. `222`
C. `263`
D. `50`
E. `321`

```
Stack<Integer> s;
s = new Stack<Integer>();
s.push(212);
s.push(43);
s.push(17);
s.pop();
s.push(-9);
s.push(8);
s.pop();
s.push(-24);
s.push(10);
s.push(56);
s.pop();
s.pop();
```

QUESTION 39

If a given Binary Search algorithm (assume average-case) takes 2 seconds to search a list of 32,000 elements, how long would it take to search a list of 1,024,000 elements?

QUESTION 40

What is the best case runtime of inserting an item into a linked list?