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.

For more Computer Science practice tests and materials,

go to www.apluscompsci.com

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)

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 $\texttt{2f}_{16}$ + $\texttt{72}_{8}$?

A. $\texttt{1231}_{4}$ B. $\texttt{1100101}_{2}$ C. $\texttt{103}_{10}$ D. $\texttt{153}_{8}$ E. $\texttt{69}_{16}$

QUESTION 2

What is output by the code to the right?

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

```
out.print(1 + 2 * 3 - 4);
```

QUESTION 3

What is output by the code to the right?

A. `31234567.13`

B. `31,234,567.13`

C. `31234567.12653`

D. `31,234,567.12653`

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

```
out.printf("%,.2f", 31234567.12653);
```

QUESTION 4

What is output by the code to the right?

A. `b4`

B. `102`

C. `true4`

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

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

```
out.print("believer".charAt(0) + 4);
```

QUESTION 5

What is output by the code to the right?

A. `true` B. `false`

```
boolean a = true;
boolean b = !a;
out.print((a)? !b&&a : b);
```

QUESTION 6

What is output by the code to the right?

A. `64.0` B. `16.0`

C. `8.0` D. `4.0`

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

```
out.print(Math.cbrt(64));
```

QUESTION 7

What is output by the code to the right?

A. `23A4.1oA4.1` B. `884.1oA4.1`

C. `92.1o69.1`

D. `92.1oA4.1`

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

```
int a = 23;
char b = 'A';
String s = "o";
double d = 4.1;
out.print(a + b + d + s + b + d);
```

QUESTION 8

What is output by the code to the right?

A. `[2, 4, 7, 11, 12]`

B. `[0, 1, 3, 6, 7]`

C. `[1, 2, 4, 6, 7]`

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

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

```
int [] borb = new int [5];
for(int i= 0; i < 5; i++) {
  borb[i] = i + 1;
}
for(int i= 0; i < 5; i++) {
  borb[i] += borb[2*i/3];
}
out.print(Arrays.toString(borb));
```

QUESTION 9

What is output by the code to the right?

A. `014916`　　B. `1491625`

C. `23236`　　D. `00220`

E. `23016`

```
for(int i = 0; i < 5; i++) {
  out.print(i^2);
}
```

QUESTION 10

What is the output by the code to the right?

A. `0 5`　　B. `0 10`

C. `15 18`　　D. `10 18`

E. Output cannot be determined until runtime.

```
int [] jeff = {1, 5, 7, 2, 3};
int r = 0; int j = 0;
for(int i = 0; i < 5; i++) {
  r += i;
  j += jeff[i];
}
out.print(r + " " + j);
```

QUESTION 11

What is the output by the code to the right?

A. `Mult1ple`　　B. `s1ices`

C. `of`　　D. `cheesecake.`

E. `cheesecake`

```
Scanner file;
file = new Scanner("Mult1ple s1ices
of cheesecake.");
String s = file.nextLine();
out.print(s.split("\\W")[3]);
```

QUESTION 12

What is the output by the code to the right?

A. `4 24 288`

B. `2 8 64`

C. `6 19 45`

D. `10 20 40`

E. There is an infinite loop.

```
int i = 1; int j = 1; int k = 1;
while(i < 10 && j < 20 && k < 40) {
  i++;
  j*= i;
  k*= j;
}
out.println(i + " " + j + " " + k);
```

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. `%(Modulus)`
II. `++(Unary post-increment)`
III. `++(Unary pre-increment)`

QUESTION 14

What is the output by the code to the right?

A. `~29` B. `29`

C. `-30` D. `-29`

E. `30`

```
out.print(~29);
```

QUESTION 15

What is the output by the code to the right?

A. `a`

B. `b`

C. `c`

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

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

```
ArrayList<String> ar;
ar = new ArrayList<String>();
ar.add("a");
ar.add("b");
ar.add("c");
ar.addAll(ar);
out.print(ar.get(4));
```

QUESTION 16

What is the output by the code to the right?

A. `15` B. `39` C. `27` D. `1` E. `9`

```
out.print(3 * 2 >> 1 + 5 ^ 9);
```

QUESTION 17

What is the output by the code to the right?

A. `true` B. `false`

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

QUESTION 18

What is the output by the code to the right?

A. `abcde` B. `acbed`

C. `debca` D. `edcba`

E. `a`

```
ArrayList<String> ar;
ar = new ArrayList<String>();
ar.add("a");ar.add("c");ar.add("b");
ar.add("e");ar.add("d");
Iterator it = ar.iterator();
while(it.hasNext()) {
  out.print(it.next());
}
```

QUESTION 19

What is the output by the code to the right?

A. `256432` B. `234652` C. `5683` D. `5899632`

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

```
int a = 234652; int b = 0;
while(a>0) {
  b*=10;
  b+=a%100;
  a/=10;
}
out.print(b);
```

QUESTION 20

What is the output by the code to the right?

A. `200219202`

B. `cehsee`

C. `eseehc`

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

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

```
String s = "cheese";
String b = "";
while(s.length()>0) {
  b+= s.charAt(0) +
            s.charAt(s.length()-1);
  s = s.substring(1, s.length()-1);
}
out.print(b);
```

QUESTION 21

What is the output of the function call `banana(3, 4)`?

A. 34   B. 43   C. 3   D. 4   E. 6

QUESTION 22

What is the output of the line marked **//line 1**?

A. 162734

B. 437261

C. 43

D. 34

E. 612734

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

public static int banana(int a,
                          int b)
{
  if(a > b) return a*10 + b;
  if(a < b) return b*10 + a;
  return a*20;
}

/////////CLIENT CODE////////
int [] bb = {1, 6, 2, 7, 3, 4};
for(int i = 1; i <bb.length ; i++) {
  bb[i] = banana(bb[i-1], bb[i]);
}
out.print(bb[5]);//line 1
```

QUESTION 23

What is the output of the line marked **//line 1**?

A. 0

B. 1

C. 2

D. Output cannot be determined until runtime.

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

QUESTION 24

What is the output of the line marked **//line 2**?

A. 4

B. 5

C. Output cannot be determined until runtime.

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

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

```
// Use the code below to answer
// questions 23-24.

abstract class wooly{
  static int bob;
  public wooly() {
     bob++;
  }
  public abstract String name();
  public int feet() {
     return 4;
  }
}

class mammoth extends wooly{
  public String name() {
     return "manny";
  }
  public int feet() {
     return 5;
  }
}

//////CLIENT CODE//////
wooly drake = new mammoth();
wooly george = new mammoth();
out.println(drake.bob); //line 1
out.println(drake.feet()); //line 2
```

QUESTION 25

What is the output by the code to the right?

A. `feg`

B. `acdefg`

C. `fedcbacdefg`

D. `fgefdecdbca`

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

```
String [] s = {"a", "b", "c", "d",
                      "e", "f", "g"};
for(int i = 1; i < 6; i++) {
  s[i] += s[i-1]+s[i+1];
}
out.print(s[5]);
```

QUESTION 26

What is the output by the code to the right?

A. `16`

B. `13`

C. `36`

D. `18`

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

```
int[] a = {1, 3, 5, 7, 0, 2, 4, 6, 8};
int[] b = {2, 6, 8, 3, 1, 0, 7, 4, 5};
for(int i = 0; i < 9; i++) {
  a[b[i]] += b[i];
}
out.print(a[8]);
```

QUESTION 27

What is the output of the function call `rep(0, 4, "barbs")`?

A. `0` B. `4` C. `10` D. `8`

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

QUESTION 28

What is the output of the function call `rep(0, 7, "superbad")`?

A. `28`

B. `22`

C. `0`

D. `18`

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

```
public static int rep(int a, int b,
String s) {
  if(a < s.length() && a>-1 &&
b<s.length() && b>-1) {
     if(s.charAt(a)>s.charAt(b))
return a + rep(a + 1, b, s);
     if(s.charAt(a)<s.charAt(b))
return b + rep(a, b - 1, s);
  }
  return 0;
}
```

QUESTION 29

What is output by the code to the right?

A. `7` B. `15` C. `31` D. `63` E. `3`

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

QUESTION 30

What could replace **<code 1>** so that an argument is added to the map without error?

A. `1, 2`

B. `'A', 4`

C. `'A' + 0, 4`

D. All of the above.

E. Both A and C.

```
TreeMap<Integer, Integer> tm;
tm = new TreeMap();
tm.put( <code 1> );
```

QUESTION 31

What is output by the code to the right?

A. `163`

B. `99`

C. `57`

D. `64`

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

```
int[][] mat = new int[5][5];

Arrays.fill(mat[0], 1);

for(int i = 1; i < 5; i++) {
  mat[i][0] = mat[i-1][0]*2;
  for(int j = 1; j < 5; j++) {
     mat[i][j] += mat[i-1][j] +
                         mat[i][j-1];
  }
}
out.print(mat[4][4]);
```

QUESTION 32

What is output by the code marked **`//1`**?

A. `string`

B. `true`

C. `false`

D. `1`

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

QUESTION 33

What is output by the code marked **`//2`**?

A. `43`

B. `true`

C. `false`

D. `0`

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

```
Stack<Object> s = new Stack();
s.add(43);
s.add("string");
s.add('c');

out.println(s.remove("string")); //1

out.println(s.remove(43)); //2
```

QUESTION 34

What is output by the code to the right?

A. `true` B. `false`

```
String s;
s = "boats and beans";
String r;
r = "\\w+\\s\\w+\\s\\w+";
out.println(s.matches(r));
```

QUESTION 35

What is output by the code to the right?

A. `13` B. `112` C. `12` D. `16`

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

```
out.print(4*3^7&4<<1+1);
```

QUESTION 36

What is the worst-case time complexity for insertion on a hash table list 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. `A ^ !B`

C. `!A ^ !B`

D. `B || A`

E. `A && !B`

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

QUESTION 38

What is the girth of the graph to the right?

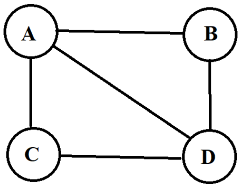
A. `3`

B. `4`

C. `5`

D. `10`

E. `Infinite`



QUESTION 39

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

```
2 3 * 8 3 - 6 3 - 4 * + +
```

QUESTION 40

What is the 8-bit binary 2’s compliment notation of the following integer?

121