QUESTION 1

What is the sum of $3E6_{16}$ and 444_{16}?

A. 720_{16} B. 777_{16} C. $7IA_{16}$ D. $83A_{16}$ E. $82A_{16}$

QUESTION 2

Which of these declares a variable to have the type `MyClass` and initializes it using the two parameter constructor with parameters `3` and `9`?

A. `(MyClass)m = MyClass(3,9)`

B. `new MyClass m(3,9)`

C. `MyClass m = 3, 9`

D. `MyClass m = new MyClass(3,9)`

E. `MyClass m = MyClass(3,9)`

QUESTION 3

If a `MyClass` object is built using the second constructor, to what is its data member `ydata` initialized?

A. Same as `xdata` B. `65535`

C. `0` D. Not initialized

E. Syntax error to not initialize

```
public class MyClass {
  public MyClass(int x, int y) {
    xdata = x; ydata = y;
  }

  public MyClass(int x) {
    xdata = x;
  }

  // methods not shown

  private int xdata, ydata;
}
```

QUESTION 4

How many `*`'s are output by the code to the right?

A. `160` B. `40` C. `26`

D. `10` E. `0`

```
for (int i=0; i<10; ++i)
  for (int j=1; j<16; j*=2)
    System.out.print("*");
```

QUESTION 5

What replaces **<*1>** in the code to the right to give the character at position `i` in `s`, converted to lower case?

A.
```
if (s.charAt(i) >= 'A' &&
    s.charAt(i) <= 'Z')
      s.charAt(i) - 'A'
```

B. `Character.isLowerCase?s[i]:s[i]-'a'`

C. `Character.toLowerCase(s[i])`

D. `Character.toLowerCase(s.charAt(i))`

E. More than one of these

```
public static void smallOutput(String s) {
  for (int i=0; i<s.length(); ++i)
    System.out.print(<*1>);
}
```

QUESTION 6

What does `int[][] a` look like after the static method call `process(a)` when `a` begins as the matrix below?

1	4	7
2	5	8
3	6	9

A.

1	2	3
4	5	6
7	8	9

B.

1	5	9
3	7	11
5	9	13

C.

1	4	7
3	7	11
7	12	13

D.

1	3	5
4	7	9
7	12	13

E.

0	1	2
1	2	3
2	3	4

```
public static void process(int[][] a) {
  for (int i=0; i<a.length; ++i)
    for (int j=0; j<a[i].length; ++j)
      a[i][j] = a[j][i] + i + j;
}
```

QUESTION 7

Which of these can replace **<*1>** in the code to the right to give a valid initialization to `b`?

A. `17`
B. `x`
C. `(boolean)x`
D. `boolean(x)`
E. None of these

```
int x = 17;
boolean b = <*1>;
```

QUESTION 8

Which of these statements sets `b` to `true`?

A. `b = (x < y) && !(z < y);`
B. `b = !(x < y) || !(y < z);`
C. `b = !( (x > y) || (y < z) );`
D. `b = !( !(x > y) && (y < z) );`
E. `b = x < y < z;`

```
int x = 1, y = 2, z = 3;
boolean b;
```

QUESTION 9

What is returned by `f("UIL")`?

A. `38` B. `39` C. `40`

D. `41` E. `42`

```
public static int f(String s) {
  int x = 0, y = 0;
  while (x < s.length()) {
    y += s.charAt(x) - 'A';
    ++x;
  }
  return y;
}
```

QUESTION 10

Which of these data structures provides insertion access only at one end of a list and removal access only at the other end?

A. `queue` B. `stack` C. `heap` D. `tree` E. `map`

QUESTION 11

What replaces **<*1>** so that the `Scanner` will use all non-digits to delimit tokens?

A. `"0-9"`

B. `"^0-9"`

C. `"[^0-9]"`

D. `"^0-^9"`

E. `"-[0-9]"`

```
Scanner s = new Scanner(System.in);
s.useDelimiter(<*1>);
```

QUESTION 12

What is output by the code to the right?

A. `22` B. `31`

C. `23` D. `4`

E. Nothing

```
int[] a = {1, 2, 3, 4, 5};
int i = 0;
int j = i++;
System.out.print(""+a[++i]+a[j++]);
```

QUESTION 13

What replaces **<*1>** in the code to the right to add the number `27` to `a`?

A. `27` B. `(Integer)27`

C. `new Integer(27)` D. Either A or C

E. A, B, or C

```
ArrayList<Integer> a =
                    new ArrayList<Integer>();
a.add(<*1>);
```

QUESTION 14

What is output by the code to the right?

A. `hello` B. `goodbye`

C. `false` D. `true`

E. Nothing

```
Set<String> s = new TreeSet<String>();
s.add("hello");
s.add("hello");
s.add("goodbye");
s.remove("hello");
System.out.print(s.contains("hello"));
```

QUESTION 15

What replaces **<*1>** in the code to the right to call the other static method `mystery()` with the parameters `a`, `rank`, the index of the first element of `a`, and one more than the index of the last element of `a`?

A. `mystery(a,rank,1,length)`

B. `mystery(a,rank,1,100)`

C. `mystery(a,rank,1,a.length())`

D. `mystery(a,rank,0,a.length())`

E. `mystery(a,rank,0,a.length)`

QUESTION 16

Assume **<*1>** is filled in correctly. What does `mystery()` do?

A. Returns the value corresponding to position `rank` in `a` if the elements of `a` were sorted without sorting all of `a`

B. Sorts `a` and returns the value corresponding to position `rank`

C. Looks for a position in `a` corresponding to value `rank`

D. Returns the median element of `a`

E. Returns the sum of the elements of `a` except for the element at position `rank`

QUESTION 17

What is the expected running time of `mystery(a,0)` when `a` is a random array with `n` elements?

A. `O(1)` B. `O(log n)` C. `O(n)`

D. $O(n^2)$ E. `O(n log n)`

```
public static void swap(int[] a, int i,
                                    int j) {
  int temp = a[i];
  a[i] = a[j];
  a[j] = temp;
}

public static int partition(int[] a,
                     int front, int back) {
  int pivot = a[front];
  int i = front+1;
  int j = back-1;
  while (i<j) {
    while((i<back) && (a[i] < pivot)) ++i;
    while((j>front) && (a[j] >= pivot))
                                        --j;
    if (i<j) swap(a,i,j);
  }
  swap(a,front,j);
  return j;
}

public static int mystery(int[] a,
                                  int rank) {
  return <*1>;
}

public static int mystery(int[] a,
           int rank, int front, int back) {
  if ((front > rank) || (back <= rank))
    throw new IllegalArgumentException();
  int mid = partition(a,front,back);
  if (mid == rank)
    return a[mid];
  else if (mid > rank)
    return mystery(a,rank,front,mid);
  else
    return mystery(a,rank,mid+1,back);
}
```

QUESTION 18

What is output by the code to the right?

A. `scienc ecomputer`

B. `scienceco mputer`

C. `computer science`

D. `computers cience`

E. Nothing

```
String s1 = "computer", s2 = "science";
StringBuffer sb = new StringBuffer(s1);
sb.append(s2);
sb.insert(9," ");
System.out.print(sb);
```

QUESTION 19

What replaces **<*1>** in the code to the right to indicate that `numPairs` is a class variable, with one value shared by all instances of `Shoe`?

A. `void`
B. `static`
C. `unique`
D. `shared`
E. `catch`

For the remaining questions, assume that **<*1>** has been filled in correctly.

QUESTION 20

What is output by the code below?

```
Shoe bigShoe = new Shoe("loafer",
                  "brown",15, "EEE");
System.out.print(bigShoe);
```

A. `brown loafer size: 15EEE`
B. `loafer brown size:15EEE`
C. `brownloafer size:15EEE`
D. The memory address of `bigShoe`
E. An exception is thrown

QUESTION 21

What is output by the code below?

```
Shoe otherShoe = new Shoe();
System.out.print(otherShoe);
```

A. `  size:`
B. `color type size: sizewidth`
C. At compile time the call to `Shoe()` will be an error
D. At run time the call to `Shoe()` will throw an exception
E. At run time the call to `print()` will throw an exception

QUESTION 22

What is the superclass of `Shoe`?

A. `String`
B. `int`
C. `Object`
D. `Clothing`
E. It has no superclass

```
public class Shoe {
  public Shoe(String type, String color,
                  int size, String width) {
    this.type = type;
    this.color = color;
    this.size = size;
    this.width = width;
    ++numPairs;
  }

  public String toString() {
    return color + " " + type + " size: "
         + size + width;
  }

  private String type, color;
  private int size;
  private String width;
  private <*1> int numPairs = 0;
}
```

QUESTION 23

Which of these declares and creates an object that associates `Book` objects with `Person` objects?

A.
```
List<Book,Person> obj =
        new ArrayList<Book,Person>();
```

B.
```
Map<Book,Person> obj =
          new TreeMap<Book,Person>();
```

C.
```
List<Book,Person> obj =
              new List<Book,Person>();
```

D.
```
Map<Book,Person> obj =
              new Map<Book,Person>();
```

E. Either A or C

QUESTION 24

Assume that the object in the previous question has been created and that items have been added to it. What type of object is returned by `obj.iterator().next()`?

A. `Book` B. `Person`

C. `Map.Entry` D. `Iterator`

E. There is no `iterator()` method for `obj`

```
public class Person {
  // constructors and methods not shown
  private String familyName, givenName;
}

public class Book {
  // constructors and methods not shown
  private String title, genre;
  private int numPages;
}
```

QUESTION 25

Which of these best describes the value `z` after executing the code to the right? (Assume the input actually contains a floating point value.)

A. The largest possible long value

B. The smallest possible long value

C. The largest integer which is not more than the floating point value read from the input

D. The smallest integer which is at least as big as the floating point value read from the input

E. The integer closest to the floating point value read from the input

```
Scanner s = new Scanner(System.in);
long z = Math.ceil(s.nextDouble());
```

QUESTION 26

What is output by the code to the right?

A. `123` B. `456`

C. `a` D. `abc`

E. Nothing

```
String[] strings =
          "abc123def456ghi".split("[a-z]");
System.out.print(strings[1]);
```

QUESTION 27

What is output by the code to the right?

A. 96 B. 37 C. 36

D. 33 E. 30

```
int x = 37, y = 59;
System.out.print(x^y);
```

QUESTION 28

What does `a` look like at the point in the execution of `sort(a)` when `i` is incremented in the `for` loop from `1` to `2` when `a` begins as the array below?

14	-2	-5	81	-8	19

A.

14	-2	-5	81	-8	19

B.

-8	-2	-5	81	14	19

C.

-8	-5	-2	81	14	19

D.

81	19	-5	14	-8	-2

E.

81	19	14	-2	-5	-8

```
public static void sort(int[] a) {
  for (int i=0; i<a.length-1; ++i) {
    int item = a[i];
    int index = i;
    for (int j=i; j<a.length; ++j)
      if (a[j] > item) {
        item = a[j];
        index = j;
      }
    int temp = a[i];
    a[i] = a[index];
    a[index]=temp;
  }
}
```

QUESTION 29

What sorting algorithm is implemented by `sort()`?

A. Selection sort B. Insertion sort

C. Quicksort D. Merge Sort

E. Random sort

QUESTION 30

Which of the following correctly declares and initializes an iterator for `list` that can be used to both to add and remove items from the list?

A.
```
ListIterator<String> iter =
                        list.iterator();
```

B.
```
ListIterator<String> iter =
                    list.listIterator();
```

C.
```
Iterator<String> iter =
                    list.listIterator();
```

D. Either A or B

E. A, B, or C

```
LinkedList<String> list =
                    new LinkedList<String>();
// code to initialize list not shown
```

QUESTION 31

The heap data structure uses an `ArrayList` to hold the elements of a complete binary tree. The root is in position `0`, its children are in positions `1` and `2`, the next level is in positions `3`, `4`, `5`, and `6`, and so on. What replaces **<*1>**, **<*2>**, and **<*3>** in the code to the right to give the index of a node's left child, right child, and parent, respectively?

A. **<*1>:** `0`
<*2>: `1`
<*3>: `2`

B. **<*1>:** `2*index`
<*2>: `2*(index+1)`
<*3>: `index/2`

C. **<*1>:** `2*index + 1`
<*2>: `2*index + 2`
<*3>: `(index-1)/2`

D. **<*1>:** `2*index`
<*2>: `2*index + 1`
<*3>: `index/2`

E. **<*1>:** `2*index - 1`
<*2>: `2*index`
<*3>: `(index+1)/2`

QUESTION 32

What property is maintained by the `Heap` class?

A. The smallest value is at the root

B. The largest value is at the root

C. All subtrees have the heap property

D. Both A and C

E. Both B and C

QUESTION 33

What is the running time of removing an item from a heap containing `n` elements?

A. `O(1)` B. `O(log n)` C. `O(n)`

D. $O(n^2)$ E. `O(n log n)`

QUESTION 34

Which of these classes could be used as elements of a `Heap` object?

A. `String` B. `Double`

C. `Random` D. Both A and B

E. A, B, and C

```
public class Heap <E extends Comparable> {
  public void add(E item) {
    data.add(item);
    int index = data.size()-1;
    while ( (index != 0) &&
            (data.get(parent(index)).
                       compareTo(item)<0) ) {
      data.set(index,
                    data.get(parent(index)));
      data.set(parent(index), item);
      index = parent(index);
    }
  }

  public E remove() {
    E item = data.get(0);
    data.set(0,
                data.remove(data.size()-1));
    int index = 0;
    while (left(index) < data.size()) {
      int left = left(index),
          right = right(index);
      int swapindex;
      if (right>=data.size())
        swapindex=left;
      else if (data.get(left).compareTo(
                          data.get(right))>0)
        swapindex=left;
      else swapindex=right;
      if (data.get(index).compareTo(
                    data.get(swapindex))<0) {
        E temp = data.get(index);
        data.set(index,
                        data.get(swapindex));
        data.set(swapindex, temp);
        index = swapindex;
      }
      else index = data.size();
    }
    return item;
  }

  private static int left(int index) {
    return <*1>;
  }
  private static int right(int index) {
    return <*2>;
  }

  private static int parent(int index) {
    return <*3>;
  }

  private ArrayList<E> data =
                         new ArrayList<E>();
}
```

QUESTION 35

What is returned by `f(0,4)`?

A. `4` B. `5` C. `6`

D. `10` E. `100`

QUESTION 36

If `x` is a positive integer, what is returned by `f(2,x)`?

A. Three more than twice `x`

B. Two more than three times `x`

C. The square of `x`

D. Two more than the square of `x`

E. One less than the square of `(x+1)`

```
public static int f(int x, int y) {
  if (x==0) return 1+y;
  if (y==0) return f(x-1,1);
  return f(x-1, f(x,y-1));
}
```

QUESTION 37

What is output by the code to the right?

A. `3.1416` B. `3.14159265`

C. `3.141` D. `3.142`

E. `3.14`

```
double pi = 3.14159265;
System.out.printf("%2.4f", pi);
```

QUESTION 38

Which of these statements is true about `process()`?

A. It can throw `IOException` and its subclasses

B. It can throw `RunTimeException` and its subclasses

C. It can throw `Error` and its subclasses

D. Both A and B

E. A, B, and C

```
public static int process(String s)
                          throws IOException {
  // code not shown
}
```

QUESTION 39

What is returned by `f("abacab")`?

A. `10` B. `12`

C. `14` D. `16`

E. `18`

```
public static int f(String s) {
  int answer = 0;
  for (int i=0; i<s.length(); ++i)
    switch((s.charAt(i)-'a')%2) {
    case 0 : ++answer; break;
    case 1 : answer *= 2; break;
    }
  return answer;
}
```

QUESTION 40

Which of these data structures uses a function that maps an object to its position in an array?

A. queue B. binary tree C. linked list D. hash table E. stack