QUESTION 1

What is the value of $267_8 + 104_8$?

A. 373_8 B. 374_8 C. 371_8 D. $36A_8$ E. 163_8

QUESTION 2

What is the value of `z` after executing the code to the right?

A. 10 B. 300 C. 1464

D. 70 E. 264

```
int x = 10;

int y = x * 4 + 6;

int z = y + 4 * 6;
```

QUESTION 3

What are the values of `ch` and `total` after executing the code to the right?

A. `ch: 'a'` `total: 25`
B. `ch: 'b'` `total: 24`
C. `ch: 'a'` `total: 26`
D. `ch: 'b'` `total: 25`
E. `ch: 'c'` `total: 24`

```
int total = 0;
char ch = 'z';

while (ch >= 'b') {
  ++total;
  --ch;
}
```

QUESTION 4

Which of these can be used inside any class to output data member `x` from `MyClass m`?

A. `System.out.print(m(x))`

B. `System.out.print(m.getX())`

C. `System.out.print(m/x)`

D. `System.out.print(m.x)`

E. Both B and D

QUESTION 5

Which of these builds a `MyClass` object with data member `x` initialized to `0.7` and data member `y` initialized to `12`?

A. `MyClass m = (0.7, 12)`

B. `MyClass m = {0.7, 12}`

C. `MyClass m(0.7, 12)`

D. `MyClass(0.7, 12)`

E. `new MyClass(0.7, 12)`

```
public class MyClass {
  public MyClass(double xInit, int yInit) {
    x = xInit;
    y = yInit;
  }

  public double getX() { return x; }
  public int getY() { return y; }

  private double x;
  public int y;
}
```

QUESTION 6

Which of these subtracts `1` from `int x` and returns the previous value?

A. `x-1?x` B. `x-1` C. `x--` D. `--x` E. Both A and D

QUESTION 7

What is output by the code to the right?

A. `e` B. `a` C. `eta`

D. `t` E. nothing

```
String s = "alpha-beta";

System.out.print(s.charAt(8));
```

QUESTION 8

What replaces **<*1>** in the code to the right to give the character at index `i` in `String s`?

A. `s+i` B. `s[i]`

C. `s.index(i)` D. `s.charAt(i)`

E. Both B and D

```
public static int compute(String s) {
  int total = 0;
  for (int i=0; i < s.length(); ++i) {
    char c = <*1>;
    if (c >= '0' && c <= '9')
      total = total + (c - '0');
  }
  return total;
}
```

QUESTION 9

Assume **<*1>** is filled in correctly. What is returned by `compute("1999")`?

A. `28` B. `1` C. `9`

D. `729` E. `2006`

QUESTION 10

What is the value of `total` after executing the code to the right?

A. `-2` B. `-1` C. `0`

D. `1` E. `2`

```
int x = 42, total = 0;
while (x != 0) {
  switch(x%3) {
  case 0: total = total + 1; break;
  case 2: total = total - 1; break;
  }
  x = x/3;
}
```

QUESTION 11

What is returned by `process(3)`?

A. `"abaaba"` B. `"abababb"`

C. `"aaaaaa"` D. `"babbab"`

E. `"bbb bbb"`

```
public static String process(int x) {
  if (x == 0)
    return "";
  else if (x<0)
    return "a" + process(-x-1) + "a";
  else
    return "b" + process(-x+1) + "b";
}
```

QUESTION 12

How many `a`'s are in the string returned by `process(-1001)`?

A. `500` B. `501` C. `1000`

D. `1001` E. `1002`

QUESTION 13

Can these static methods all be part of the same class?

A. Yes

B. No, because the names of the methods are the same

C. No, because the first and second method have the same name, return type, and parameter names

D. No, because the methods have the same name and they all have two parameters

E. No, because the first and third method have the same name and parameter types

```
public static int f(int x, int y) {
  // code not shown
}

public static int f(int x, double y) {
  // code not shown
}

public static double f(Integer x,
                       Integer y) {
  // code not shown
}
```

QUESTION 14

Which of these replaces **<*1>** in the code to the right to create arrays that have only one copy for the class, not one for each instance, and that are accessible only inside the `Date` class?

A. `int[12]`

B. `private int[]`

C. `private static int[]`

D. `static int[12]`

E. More than one of these

QUESTION 15

Assume **<*1>** is filled in correctly. Which of these causes an `IllegalArgumentException` to be thrown?

A. `new Date(31,7,2000)`

B. `new Date(31,7,2100)`

C. `new Date(29,2,2000)`

D. `new Date(29,2,2100)`

E. More than one of these

```
public class Date {
  public Date(int day, int month, int year)
  {
    if (year <= 0)
      throw new IllegalArgumentException(
                  "Bad year");
    this.year = year;

    if (month <= 0 || month > 12)
      throw new IllegalArgumentException(
                  "Bad month");

    if (isLeapYear(year)) {
      this.dayofyear =
                  leapoffset[month-1]+day;
      if (dayofyear>leapoffset[month])
      throw new IllegalArgumentException(
                  "Bad day");
    }
    else {
      this.dayofyear = offset[month-1]+day;
      if (dayofyear>offset[month])
      throw new IllegalArgumentException(
                  "Bad day");
    }
}

  private static boolean isLeapYear(int y)
  {
    if (y%400 == 0) return true;
    if (y%100 == 0) return false;
    if (y%4 == 0) return true;
    return false;
  }

  // other methods not shown

  private int dayofyear, year;
  <*1> offset = {0,31,59,90,120,151,181,
                 212,243,273,304,334,365};
  <*1> leapoffset ={0,31,60,91,121,152,182,
                 213,244,274,305,335,366};

}
```

QUESTION 16

What replaces **<*1>** in the code to the right to remove the first element from `elements` and return its value?

A. `elements.removeFirst(); return;`

B. `return elements.removeFirst();`

C. `return elements.remove(length);`

D. `return elements.remove(1);`

E. `return elements.dequeue();`

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

QUESTION 17

What is output by static method `test()`?

A. 312 B. 321 C. 231

D. 123 E. 132

QUESTION 18

What is the total running time of using the `enqueue()` method to insert `n` items? Choose the most restrictive correct answer.

A. $O(1)$ B. $O(\log n)$ C. $O(n)$

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

```
public class MyQueue<E> {

  public MyQueue() {
    elements = new LinkedList<E>();
  }

  public void enqueue(E element) {
    elements.addLast(element);
  }

  public E dequeue() {
    <*1>
  }

  public boolean isEmpty() {
    return elements.size() == 0;
  }

  private LinkedList<E> elements;

  public static void test() {
    MyQueue<Integer> testQ =
      new MyQueue<Integer>();
    testQ.enqueue(1);
    testQ.enqueue(2);
    testQ.enqueue(3);
    System.out.print(testQ.dequeue());
    System.out.print(testQ.dequeue());
    System.out.print(testQ.dequeue());
  }


}
```

QUESTION 19

On which of these arrays would static method `process()` return `3` if the array was passed as its parameter?

A.

1	2	3	4	5	6

B.

6	5	4	3	2	1

C.

1	3	5	2	4	6

D.

2	4	6	1	3	5

E.

1	3	5	6	4	2

```
public static int process(int[] array) {
  int returnValue = 0;
  for (int i=array.length; i>0; --i)
    if (array[i]>array[i-1])
      ++returnValue;
  return returnValue;
}
```

QUESTION 20

What replaces **<*1>** and **<*2>** in the code to the right so that it is syntactically correct?

A. **<*1>:** `abstract class`
   <*2>: `extends`

B. **<*1>:** `interface`
   <*2>: `extends`

C. **<*1>:** `abstract class`
   <*2>: `implements`

D. **<*1>:** `interface`
   <*2>: `implements`

E. Either A or D

```
public <*1> Shape {
  public abstract double area();
  public abstract double perimeter();
}

public class Circle <*2> Shape {
  public Circle(double radius) {
    r = radius;
  }
  public double area() {
    return Math.PI * r * r;
  }
  public double perimeter() {
    return 2 * Math.PI * r;
  }
  private double r;
}
```

QUESTION 21

Assume **<*1>** and **<*2>** are filled in correctly. Which of these can be used in any class to output the area of `Circle c`?

A. `System.out.print(c.area())`

B. `System.out.print(area(c))`

C. `System.out.print(c.area(r))`

D. `System.out.print(Math.PI*c.r*c.r)`

E. All of these

QUESTION 22

What is output by the code to the right?

A. `123458` B. `1234589`

C. `123459` D. `12345589`

E. The code contains an infinite loop

```
int [] intArray = {1, 2, 3, 4, 5};
int i = 0;
try {
  while (true) {
    System.out.print(intArray[i]);
    ++i;
  }
}
catch(Exception e) {
  System.out.print(8);
}
finally {
  System.out.print(9);
}
```

QUESTION 23

What is output by the code to the right?

A. `89` B. `19`

C. `98` D. `99`

E. `9`

```
int x = 3^2;
int y = (int)Math.pow(3.0,2.0);

System.out.print("" + x + y);
```

QUESTION 24

Which of these input strings would cause the code to the right to set `x` to `12`, `y` to `0.2`, and `s` to `" 19"`?

A. `120.219` B. `12 0.2 19`
C. `12.2 19` D. `12.219`
E. Both B and C

```
Scanner input = new Scanner(System.in);

int x = input.nextInt();

double y = input.nextDouble();

String s = input.nextLine();
```

QUESTION 25

What is output by the code to the right?

A. `catfalse` B. `cattrue`
C. `catchfalse` D. `cat0`
E. An exception is thrown

```
String s = "cat", t = "catch";

System.out.print(t.substring(0,3));

System.out.print(s.equals(
                        t.substring(0,3)));
```

QUESTION 26

What replaces **<*1>** to call the other `MyClass` constructor with the parameter `0`?

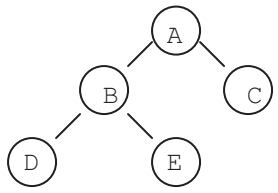
A. `0;` B. `MyClass(0);`
C. `this(0);` D. `super(0);`
E. `MySuperClass(0);`

```
public class MyClass extends MySuperClass {
  public MyClass() {
    <*1>
  }

  public MyClass(int i) {
    // code not shown
  }

  // other data and methods not shown
}
```

QUESTION 27

What order are the nodes processed in an in-order traversal of this binary tree?

A. `A B C D E` B. `E D C B A`
C. `D B E A C` D. `E D A C B`
E. `A B D E C`



QUESTION 28

What is returned by `m.keySet()`?

A. A `Set` containing `Map.Entry` objects
B. A `Set` containing `Integer` objects
C. A `Set` containing `String` objects
D. An `Iterator`
E. A `Map` from `Integer` objects to `String` objects

```
Map<Integer, String> m =
  new TreeMap<Integer, String>();
```

QUESTION 29

Which of these best describes the static method `sort()` to the right?

A. Sorts in ascending order by finding the smallest remaining item each pass of the outer loop

B. Sorts in ascending order by finding the largest remaining item each pass of the outer loop

C. Sorts in descending order by finding the smallest remaining item each pass of the outer loop

D. Sorts in descending order by finding the largest remaining item each pass of the outer loop

E. Does not modify the array

```
public static void sort(Comparable[] array)
{
  for (int i=0; i<array.length-1; ++i) {
    Comparable c = array[i];
    int cIndex = i;
    for (int j=i+1; j<array.length; ++j)
      if (c.compareTo(array[j])>0) {
        c = array[j];
        cIndex = j;
      }
    array[cIndex] = array[i];
    array[i] = c;
  }
}
```

QUESTION 30

What is the running time of `sort()` on an array containing `n` items? Choose the most restrictive correct answer.

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

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

QUESTION 31

What replaces **<*1>** in the code to the right to cause all of the items in `intArray` to be output?

A. `for (iterator<Integer> item in data)`

B. `for (item : data)`

C. `for (Integer item : data)`

D. `for (Integer item : ArrayList<Integer> data)`

E. `for (Integer item in data)`

```
ArrayList<Integer> data;
// code to initialize data not shown

<*1> {
  System.out.println(item);
}
```

QUESTION 32

What replaces **<*1>** in the code to the right to check whether the character at position `i` in `s` is upper case?

A. `Character.toUpperCase(s.charAt(i))`

B. `Character.upperCase(s.charAt(i))`

C. `Character.isUpperCase(s.charAt(i))`

D. `s.charAt(i) + 'A' - 'a'`

E. `s.charAt(i) + isLowerCase?32:0`

```
public static int numberUpper(String s) {
  int i = 0, total = 0;
  while (i < s.length()) {
    if (<*1>) ++total;
      ++i;
  }
  return total;
}
```

QUESTION 33

What replaces **<*1>** and **<*2>** in the code to the right so that all of `s` up to but not including the last space character is put in `firstName` and all of `s` after but not including the last space character is put in `lastName`? (Assume the loop to set `index` finds a space character.)

A. **<*1>:** `s.substring(0,index-1)`
<*2>: `s.substring(index)`

B. **<*1>:** `s.substring(0,index)`
<*2>: `s.substring(index-1)`

C. **<*1>:** `s.substring(0,index)`
<*2>: `s.substring(index)`

D. **<*1>:** `s.substring(0,index)`
<*2>: `s.substring(index+1)`

E. **<*1>:** `s.substring(0,index-1)`
<*2>: `s.substring(index+1)`

```
public class Employee {

  public Employee(String s, int id) {
    idNumber = id;
    int index == s.length()-1;
    while (s.charAt(index) != ' ') --index;
    firstName = <*1>;
    lastName = <*2>;
  }

  public String toString() {
    return <*3>;
  }

  private String firstName, lastName;
  private int idNumber;
}
```

QUESTION 34

Assume **<*1>** and **<*2>** are filled in correctly. What replaces **<*3>** in the code to the right to give a string containing an employee's last name, a comma, the first name, and the ID number in brackets?

A. `lastName + "," + firstName + "[" + idNumber + "]"`

B. `lastName + ',' + firstName + '[' + (String)idNumber + ']'`

C. `"lastName, firstName[idNumber]"`

D. `lastName, firstName[idNumber]`

E. `lastName + "," + firstName + "[" + (String)idNumber + "]"`

QUESTION 35

Which of these characters in a regular expression indicates that the previous group should appear zero or more times?

A. `-` B. `/` C. `.` D. `*` E. `+`

QUESTION 36

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

A. `0` B. `4` C. `5`

D. `19` E. `20`

```
int x = 10;
do {
  System.out.print('*');
  if (--x == 7) break;
  System.out.print('*');
} while (x > 0);
```

QUESTION 37

What replaces **<*1>** in the code to the right to return true if the ratio of square feet to number of rooms is smaller than `min`? Use floating point arithmetic for the division.

A. `((double)squareFeet)/numRooms < min`

B. `(double)(squareFeet/numRooms) < min`

C. `(double)(squareFeet)/numRooms < min`

D. `double(squareFeet)/numRooms < min`

E. Both A and C

```
public class Deed {

  // constructor and some methods not shown

  public boolean smallRooms(double min) {
    return <*1>;
  }

  private String address;
  private int squareFeet;
  private int numRooms;
}
```

QUESTION 38

Suppose `Deed` does not override the `toString()` method. In which class is the `toString()` method located that is called when a `Deed` object is output with `System.out.print()`?

A. `Object` B. `System`

C. `Deed` D. `IO`

E. Printing is an invalid operation on `Deed` objects

QUESTION 39

On which of these arrays does the static method call `search(array, 6)` return `4`?

A.

1	3	5	5	8	12

B.

2	3	4	5	6	7

C.

18	3	24	12	6	10

D. Both A and B

E. Both B and C

```
public static int search(int[] array,
                         int item) {
  int index = 0;
  while (array[index++] < item);
  return index;
}
```

QUESTION 40

Which of these is the binary equivalent of the hexadecimal number 19_{16}?

A. 10001_2 B. 01101_2 C. 10101_2 D. 10011_2 E. 11001_2