Sorting

An int array stores the following values. Use the array to answer the next 4 questions

9	4	12	2	6	8	18

01. Which of the following lists of numbers would accurately show the array after the first pass (outer loop) through the Selection Sort algorithm?

A. 9, 4, 12, 2, 6, 8, 18
B. 4, 9, 12, 2, 6, 8, 18
C. 2, 4, 12, 9, 6, 8, 18
D. 2, 4, 6, 8, 9, 12, 18
E. 2, 4, 9, 12, 6, 8, 18

02. Which of the following lists of numbers would accurately show the array after the fourth pass (outer loop) of the Selection Sort algorithm?

A. 9, 4, 12, 2, 6, 8, 18
B. 2, 4, 6, 9, 12, 8, 18
C. 2, 4, 6, 8, 12, 9, 18
D. 2, 4, 6, 9, 8, 12, 18
E. 2, 4, 6, 8, 12, 9, 18

03. Which of the following lists of numbers would accurately show the array after the first pass (outer loop) of the Insertion Sort algorithm?

A. 9, 4, 12, 2, 6, 8, 18
B. 2, 4, 6, 9, 12, 8, 18
C. 9, 4, 12, 2, 6, 8, 18
D. 4, 9, 12, 2, 6, 8, 18
E. 2, 4, 6, 8, 12, 9, 18

04. Which of the following lists of numbers would accurately show the array after the third pass (outer loop) of the Insertion Sort algorithm?

A. 9, 4, 12, 2, 6, 8, 18
B. 2, 4, 9, 12, 6, 8, 18
C. 2, 9, 4, 12, 6, 8, 18
D. 4, 9, 12, 2, 6, 8, 18
E. 2, 4, 6, 8, 12, 9, 18

Searching

05. Assume that a sorted array has the following values.

5	8	13	23	24	25	43	51	67	69	70	71	72

A. Using a binary search, **what values** are examined (list in order) to determine if **23** is in the array

Searching for 23: 43, 13, 24, 23

B. Using binary search**, what values** are examined (list in order) to determine if **68** is in the array?

Searching for 68: 43, 69, 51, 67

Inheritance

01. Consider the classes bellow to answer the following questions.

```
public class Animal {

  private String name;

  public Animal(String name) {
          this.name = name;
  }

  public String getName() {
          return name;
  }

  public void setName(String name) {
          this.name = name;
  }

  public void makeNoise() {
          System.out.println("grrrrrrrr");
  }

  public void eat() {
          System.out.println("eating …");

  }

  public void walk() {
          System.out.println("walkning …");

  }

}
```

```
public class Dog extends Animal {

  public Dog(String name) {
          super(name);
  }

  public void makeNoise() {
          System.out.println("woof");
  }

}
```

```
public class Cat extends Animal {

  public Cat(String name) {
          super(name);
  }

  public void makeNoise() {
          System.out.println("meow");
  }

}
```

01. Classes DOG and CAT extend from Animal (True or False)?

A. True
B. False

02. DOG and CAT are super classes while Animal is a subclass (True or False)?

A. True
B. False

03. The field name is inherited by DOG and CAT from Animal (True or False)?

A. True
B. False

04. The method eat() is inherited by DOG and CAT from Animal (True or False)?

A. True
B. False

05. The method walk() is inherited by DOG and CAT from Animal (True or False)?

A. True
B. False

```
public class Animal {

  private String name;

  public Animal(String name) {
         this.name = name;
  }

  public String getName() {
         return name;
  }

  public void setName(String name) {
         this.name = name;
  }

  public void makeNoise() {
         System.out.print("grrrrrrr");
  }

  public void eat() {
         System.out.print("eating …");

  }

  public void walk() {
         System.out.print("walkning …")
  }

}
```

```
public class Dog extends Animal {

  public Dog(String name) {
         super(name);
  }

  public void makeNoise() {
       System.out.print("woof");
  }

}
```

```
public class Cat extends Animal {

  public Cat(String name) {
         super(name);
  }

  public void makeNoise() {
       System.out.print("meow");
  }

}
```

06. The Animal method makeNoise() is overridden by the Cat's and Dog's makeNoise() methods (True or False)?

A. True
B. False

07. Write a statement to create an instance of a cat named Tom:

Cat tom = new Cat("Tom");

08. Write a statement to execute the makeNoise() method from the Animal class. You output should be "grrrrrrr"

Animal animal = new Animal("Mickey");
animal.makeNoise();

09. By making use of polymorphism, use your Animal variable created before to execute the Dog's makeNoise() method.

Animal animal = new Dog("Pluto");
animal.makeNoise();

10. What should be the output of this code?

```
Animal dogCatOthers = new Animal("Mickey");
 dogCatOthers.makeNoise();
 dogCatOthers = new Cat("Tom");
 dogCatOthers.makeNoise();
 dogCatOthers = new Dog("Pluto");
 dogCatOthers.makeNoise();
```

A. "grrrrrrrrr "
B. "woof "
C. "meow"
D. "grrrrrrrrrmeowwoof "
E. error

Recursion/Text Processing & Wrapper Classes

Consider the classes bellow to answer the following questions.

```
1  public static int arithmeticProgression (int a1, int
2                                          d, int n) {
3
4    if (n == 1) {
5        return a1;
6    } else {
7        return d + arithmeticProgression(a1, d, n-1);
8    }
9
10}
```

```
public static void main(String[] args) {

  System.out.println(arithmeticProgression(2, 3, 7));

}
```

01. Is the arithmeticProgression method correct (it finds the expected output) (True or False)?

[✓ A. True]
B. False

02. Line 5 is the recursive case (True or False)?

C. True
[✓ D. False]

03. Line 7 is the base case (True or False)?

A. True
[✓ B. False]

04. What is the depth of this recursion?

A. 3
B. 4
C. 5
[✓ D. 6]
E. 7

Consider the classes bellow to answer the following questions.

```
1 public static String charSearch(String str, char ch) {
2
3    if (str.length() == 0) {
4        return "not found";
5    }
6    else if (str.charAt(0) == ch) {
7        return "found";
8   } else {
9         return charSearch(str.substring(1), ch);
10  }
11 }
```

```
public static void main(String[] args) {

  System.out.println(charSearch("house", 'u'));

}
```

05. Is the charSearch method correct (it finds the expected output) (True or False)?

E. True
F. False

06. Line 4 is the base case (True or False)?

G. True
H. False

07. Line 7 is the base case (True or False)?

C. True
D. False

08. What is the depth of this recursion?

F. 2
G. 3
H. 4
I. 5
J. 6

09. Rewrite the method to return the number of digits found in the string str?

```
public static int searchDigit(String str) {

   if (str.length() == 0) {
       return 0;
   }
   else if (Character.isDigit(str.charAt(0))) {
       return 1 + searchDigit(str.substring(1));
  } else {
        return searchDigit(str.substring(1));
  }
}
```