Name: _____ **Class:** _____ **Date:** _____ **ID: A**

Unit 4 - Objects and Classes Questions

True/False
Indicate whether the statement is true or false.

____ 1. The name of a class in Java must be the same as the name of its source file (excluding the extension `.java`).

____ 2. The names of classes are case-sensitive.

____ 3. The `import` statement tells the compiler which other classes use this class.

____ 4. A Java program can have as many classes as necessary.

____ 5. Every class has a method called `main`.

____ 6. Fields of a class are usually declared `private`.

____ 7. An object has to be created before it can be used.

____ 8. A class may have more than one constructor.

____ 9. The programmer gives names to objects in his program.

____ 10. In Java, Integer is a class, whereas int is a primitive type.

____ 11. Instance variables that are declared public violate the principle of encapsulation.

____ 12. Every method must have a return statement.

____ 13. A constructor must have the same name as its class.

____ 14. A variable declared in one method may be used in any other method in the same class.

____ 15. Overloaded methods have a signature, which includes the number, type, and order of parameters.

____ 16. An object may be made up of other objects.

____ 17. Only one object may be created from a particular class.

____ 18. Methods that provide services to clients should be made private.

____ 19. Constructors should always return void.

____ 20. Parameters to methods may only be primitive types.

Multiple Choice
Identify the choice that best completes the statement or answers the question.

____ 21. Which statement would we use to create an object from a class called `Thing`?

a. `Thing something`
b. `Thing something = Thing( );`
c. `Thing something = new Thing;`
d. `Thing something = new Thing( );`
e. `new Thing( ) = something;`

____ 22. Suppose we have a variable something that is a reference to a `Thing` object. How would we call the method `doIt` on our `Thing` object?

a. `doIt( )`
b. `something.doIt( )`
c. `doIt(something)`
d. `something/doIt`
e. `something(doIt)`

____ 23. Object is to class as

a. circle is to square
b. house is to blueprint
c. blueprint is to house
d. bicycle is to car
e. car is to bicycle

____ 24. When a Coin object is passed to the println method.

a. a compile error occurs
b. a runtime error occurs
c. the toString method is called on the object to get the string to print
d. a default string that includes the class name is generated
e. the Coin is flipped and the result printed

____ 25. The keyword `void` is placed in front of a method name when it is declared to indicate that

a. the method does not return a value
b. the method is a constructor
c. the method is overloaded
d. the method should be called only within its class
e. the method returns a value of an unknown type

____ 26.
```
public   class  Point
{
      private  int  myX;
      private  int  myY;

      public  Point( )
      {
            myX  =  0;
            myY  =  0;
      }

      public   Point(int  x,  int  y)
      {
            myX  =  x;
            myY  =  y;
      }

      public   int   getX( )
      {
            return  myX;
      }

      public  int  getY( )
      {
            return  myY;
      }
}
```

Suppose we want to add to the `Point` class a method with the following signature.

```
// Sets the x coordinate of the point to the given value
public  void  setX(int  x)
```

Which statement should be in the body of the method?

a. `x  =  myX;`
b. `myX  =  x;`
c. `myX  =  0;`
d. `x  =  0;`
e. `myX  =  myY;`

____ 27. Consider a method that will calculate and return the sales tax on an item. Which method signature would be most appropriate?

a. `int  computeTax(double  price)`
b. `void  computeTax(double  price)`
c. `int  computeTax( )`
d. `double  computeTax(double  price)`
e. `void  computeTax( )`

____ 28.
```
public   class  Point
{
      private  int  myX;
      private  int  myY;

      public  Point( )
      {
            myX  =  0;
            myY  =  0;
      }

      public   Point(int  x,  int  y)
      {
            myX  =  x;
            myY  =  y;
      }

      public   int   getX( )
      {
            return  myX;
      }

      public  int  getY( )
      {
            return  myY;
      }
}
```

Which of the following statements creates a point with coordinate (0, 0)?

```
I.      p  =  new  Point( );
II.     p  =  Point( 0, 0);
III.    p  =  (0, 0);
```

a. I only
b. II only
c. III only
d. I and II only
e. II and III only

____ 29.
```
public   class  Point
{
      private  int  myX;
      private  int  myY;

      public  Point( )
      {
            myX  =  0;
            myY  =  0;
      }

      public   Point(int  x,  int  y)
      {
            myX  =  x;
            myY  =  y;
      }

      public   int   getX( )
      {
            return  myX;
      }

      public  int  getY( )
      {
            return  myY;
      }
}
```

Which of the following statements is true regarding the `Point` class?

a. The class won't compile because there are two methods named `Point`.
b. Variables `myX` and `myY` can be changed from outside the class.
c. `Point` objects are immutable.
d. It's impossible to create `Point` objects with coordinates other than (0, 0).
e. Giving `myX` and `myY` private visibility was a poor design decision.

Short Answer

30. Write a method header for a method named `translate` that takes an integer parameter and returns a double.

31. Write a method header for a method named `find` that takes a String and a double as parameters and returns an integer.

32. Write a method header for a method named `printAnswer` that takes three doubles as parameters and doesn't return anything.

33. Write a method called `average` that accepts two integer parameters and returns their average as a floating point value.

34. Overload the `average` method of the previous question such that if three integers are provided as parameters, the method returns the average of all three.

35. What are the differences between a class (or static) method and an instance method?
What about static fields and instance fields?
Do static methods have access to instance methods or fields?
Do instance methods have access to static methods or fields?

    To answer this question you may need to create a class with static and non-static (or instance) methods and fields and try it out.

Essay

36. Create a class `Book` with two private `int` fields, `numPages` and `currentPage`. Supply a constructor that takes one parameter and sets `numPages` to that value and `currentPage` to 1. Provide accessor methods for both fields. Also provide a method `nextPage` that increments `currentPage` by 1, but only if `currentPage` is less than `numPages`.

    Hint:

    ```
    if (currentPage < numPages)
        currentPage++;
    ```

    Create a `BookTest` class with a `main` method. Let `main` create a `Book` object with 3 pages, then call its `nextPage` method three times, printing out the value of `currentPage` after each call.