**Written Assignment 3 Writing Classes**

**CSC 123 Spring 2021**

**Short Answers. (1 points each)**

**True or False.**

1. The key word this is the name of a reference variable that an object can use to refer to itself. \_\_\_True\_\_\_\_\_
2. An instance of a class does not have to exist in order for values to be stored in a class's static fields. \_\_\_True\_\_\_\_\_
3. If you write a toString method for a class, Java will automatically call the method any time you concatenate an object of the class with a String. \_\_\_True\_\_\_\_\_
4. When an object reference is passed to a method, the method cannot change the values in the object. \_\_False\_\_\_\_\_\_

**Multiple Choice**

1. \_\_\_A\_\_\_\_\_ is the term for the relationship created by object aggregation.

|  |  |  |
| --- | --- | --- |
| A. "Has a" | C. | "Is a" |
| B. Inner class | D. | One-to-many |

1. Static methods can only operate on \_\_\_\_B\_\_\_\_ fields.

|  |  |  |
| --- | --- | --- |
| A. instance | C. | global |
| B. static | D. | local |

1. Which of the following is ***not*** true about static methods? \_\_\_C\_\_\_\_\_

|  |
| --- |
| 1. It is not necessary for an instance of the class to be created to execute a static method. |
| 1. They are called by placing the key word static after the access specifier in the method header. |
| 1. They are called from an instance of the class. |
| 1. They are often used to create utility classes that perform operations on data but have no need to collect and store data. |

1. When a reference variable is passed as an argument to a method \_\_\_B\_\_\_\_\_.

|  |
| --- |
| 1. a copy of the variable's value is passed into the method's parameter |
| 1. the method has access to the object that the variable references |
| 1. the method becomes a static method |
| 1. the program terminates |

1. You cannot use the == operator to compare the contents of \_\_\_D\_\_\_\_\_.

|  |  |  |
| --- | --- | --- |
| A. boolean variables | C. | integer variables |
| B. character variables | D. | objects |

1. If the this variable is used to call a constructor, \_\_\_A\_\_\_\_\_.

|  |
| --- |
| 1. a compiler error will result if it ***is not*** the first statement of the constructor |
| 1. a compiler error will result if it ***is*** the first statement of the constructor |
| 1. nothing will happen |
| 1. the this variable cannot be used as a constructor call |

1. Which of the following is ***not*** true about static methods? \_\_\_B\_\_\_\_\_

|  |
| --- |
| 1. They are created by placing the key word static after the access specifier in the method header. |
| 1. It is necessary for an instance of the class to be created to execute the method. |
| 1. They are called directly from the class. |
| 1. They are often used to create utility classes that perform operations on data but have no need to store and collect data. |

1. If the following is from the method section of a UML diagram, which of the statements below is true? \_\_\_\_D\_\_\_\_

+ equals(object2:Stock) : boolean

|  |
| --- |
| 1. This is a private method that returns a boolean value. |
| 1. This is a public method that returns a reference to a String object. |
| 1. This is a private method that receives two objects from the Stock class and returns a boolean value. |
| 1. This is a public method that accepts a Stock object as its argument and returns a boolean value. |

**Draw a UML Diagram for the following class: (5 points)**

public class Store

{

public double taxRate;

private String name;

//Constructor

public Store( String name )

{

this.name = name;

}

//Accessors

public String getName( )

{

return name;

}

//Mutators

public void setName( String name )

{

this.name = name;

}

//other methods

public String toString( )

{

return String.format("Name: %s\n", name );

}

public boolean equals( Object o )

{

if ( ! ( o instanceof Store ) )

return false;

else

{

Store s = (Store) o;

return ( name.equalsIgnoreCase( s.name ) );

}

}

}

Ans:

|  |
| --- |
| Store |
| + taxRate : double  - name : String |
| + Store(name : String)  + getName() : String  + setName(name : String) : void  + toString() : String  + equals(o : Object) : boolean |