### :\_\_\_\_\_\_\_\_\_\_\_\_\_

**Application Engineering and Development INFO 5100**

# Exam II

## Saturday Nov 2, 2013

1. Consider the following two variations of the order class. What is the difference between the two definitions. The “:” represent details that are not important.

class Order {

int static id = 0;

public void Order(){

:

Id = id +1;

:

}

Class Order {

int id = 0;

Public void Order(){

:

Id = id+1;

:

}

1. Companies have different departments each is responsible for certain aspects of their business. These include sales, customer service, accounting, and finance. Sales are responsible for orders, accounting is responsible for invoices, and finance is responsible for payments. Each department has a staff of people responsible for the related work. Show how to use inheritance and class hierarchy to create a reusable class and their associated subclasses. Your answer must provide detailed java code and show all relevant methods and attributes on the classes.
2. Define a complete java method [getAllProductsBelowPrice(n)] on the ProductCatalog class. The method will return the list of products with a price below a parameter n where n is an int. Assume all helper methods already exist.
3. Answer with true or false for each of the following statements:
   1. A class does not need to be an abstract class to have abstract methods defined on it.
   2. An interface class can have attributes defined as part of the class
   3. Can use the new operator on abstract classes if some of the attributes are static
   4. A variable that is declared inside a public method (local variable) is visible to all methods of the class.
   5. A local variable exists only for the duration of method execution
   6. A variable that is declared private cannot be accessed outside the class.
   7. The value stored inside a variable that is declared private can never be shared outside the class directly or indirectly.
   8. You cannot extend a class that is declared final
4. Consider the following abstract class definition of the productcatalog class. Show how to extend this class to provide an implementation for the getAllProductsBelowPrice(n) defined in question 3 above.

public abstract class AbstractProductCatalog {

String name;

ArrayList<product> products;

public AbstractProductCatalog(){

products = new ArrayList<Product>;

}

Public abstract ArrayList getProductsBelowPrice(int n);

public void setName(String n){

name = n;

}

:

:

}

.