CSE1325 OOP Midterm

July 18, 2014

Name: Key UTA ID: 100

Instructions:

1. Read all of the instructions for each question and answer what is asked. Do not write down random stuff if you don’t know the answer.
2. All questions that have the same question number are related to each other. However, they are not all necessarily dependent on each other so you can skip around if needed.
3. The test is worth 100 points and there will be 10 points extra credit available.
4. If you have a question during the test, raise your hand and ask the proctor. You may or may not get an answer, but you won’t know unless you ask.
5. Check for bonus questions.

1.a. You are being asked to write a Java system to support a veterinary hospital that treats pets, working farm animals, and animals rescued from abusive situations. For these three groups of animals, there are some similarities as well as some differences. Design a class that can be used as a superclass for all three groups of animals and give it at least 4 pieces of data. Do not include a name field in this superclass. In the style of the design documents you have turned in, define the superclass with four or more data fields, then define a subclass for each of the three groups of animals and add at least three pieces of data in each subclass that are different from each other. You do not need to define any methods. Make it clear in your design which class is the superclass for the other three. {12 points}

This is an EXAMPLE of the kind of answer that would work. You

enum ageUnit {HOURS, DAYS, WEEKS, MONTHS, YEARS};

enum animalType {PET, WORK, RESCUE}; enum workType {FARM, RANCH, ZOO, STABLE, OTHER};

public class AnimalVet //superclass public class AnimalWork //superclass

{ {

// examples of data members – no name workType working; //data member 1

double weightInLbs; //data member 1 String vetProblem; //data member 2

int age; //data member 2 Calendar problemDate; //data member 3 ageUnit ageMsmt; //data member 3 }

animalType aType; //data member 4

}

public class AnimalPet //superclass public class AnimalRescue //superclass

{ {

// examples of data members String nameTemp; //data member 1

String name; //data member 1 String species; //data member 2

String ownerLastName; //data member 2 String breed; //data member 3

String ownerFirstName; //data member 3 }

}

1.b. Think of a task that could be used by two of the three subclasses and define an interface for that task that could be used so that each subclass could define its own version of the task but using the same method name. {8 points}

public interface TrainingRecord

{

public boolean CompletedObedienceTraining

{ // for Pet and Rescue animals

}

}

2. Match the following Swing components to their definitions: {12 points}

**a**  JComboBox  **c**  JMessageDialog

**f**  JFrame  **b**  JScrollPane

**e**  JTextArea  **d**  ActionListener

a. A drop down menu of items that can have single and multiple selections

b. A window that has a scroll bar to allow user to move to text that is our of view given size of window

c. An output window which allows a message to be displayed to the user with an icon, a button to close, and the maximize, minimize, and close buttons on it

d. An interface that captures events from the user

e. A location for messages or for data to be typed in

f. A component that can have other components added to it and must be setVisible to be seen

3. Given the following Java code, answer the questions below:

enum FoodGroup { FRUITS, VEGETABLES, GRAINS, PROTEIN, DAIRY, OILS, OTHER };

public class Ingredient {

private String name;

private String faveFoodContaining;

private MetricUnitsOfMsmt units;

private FoodGroup group;

private double costPerOneUnit;

// Ingredient methods go here

}

3.a. Write a constructor method for the Ingredient class that takes in values for all data members, and write the signatures for all of the typical methods that would be needed for the class based on the data defined above. {10 points}

// constructor – correct access, name, and parameters

public Ingredient (String nm, String containing, MetricUnitsOfMsmt uni, FoodGroup grp, double cost1)

{

// must have all data members as input

// only double is possible to validate so use a set function for cost

name = nm;

faveFoodContaining = containing;

units = uni;

group = grp;

setCostPerOneUnit(cost1);

}

// set and get functions for all 5 data members – correct return types and parameters

public boolean setCostPerUnit(double c1)

public void setName(String nm)

public void setFaveFoodContaining(String containing)

public void setUnits(MetricUnitsOfMsmt uni)

public void setGroup(FoodGroup grp)

public double getCostPerUnit()

public String getName()

public String getFaveFoodContaining()

public MetricUnitsOfMsmt setUnits()

public FoodGroup setGroup()

public String toString()

3.b. Create a Java class called Dish that will contain the recipe for making the dish that is your favorite food. (Assume your favorite food requires some cooking, i.e. a raw apple would not be a favorite food for purposes of this question but apple pie could be.) The Dish class should have a list of ingredients that gives the raw ingredient name, the amount in metricUnitsOfMsmt of the raw ingredient that is in the recipe for your favorite food, and the calculated cost to purchase the raw ingredient for your favorite food. You may, if desired, create an intermediate class to represent a recipe ingredient, i.e. a class with the raw ingredient, amount needed, and calculated cost. Assume that the amounts are whatever is needed for a normal dish (or recipe) of your favorite food whether that is for one person or for multiple people. List all the data elements of the class Dish. You do not need to write the method signatures for Dish (or other classes you create) – only the data members. {6 points}

public class RecipeIngredient

{

private Ingredient rawIngredient;

private double numberOfMsmtUnits;

private double ingredientCost; // This is calculated

}

// if a totalCost is included but not individual costs -.5

public class Dish

{

private String dishName; // extra credit for including a name

private RecipeIngredient[] ingredients;

private String[] recipeSteps; // should be something about the recipe at this point

}

3.c. A useful Dish class should also have some way of representing recipe steps to take and the order of those steps. An example of a recipe step would be to "bake the mixture at 325 degrees F for 10 minutes." In your Dish class, how would you choose to represent the recipe steps needed to create and cook the Dish? Explain your representation scheme, discuss whether it fits the model of OO Design, and give at least one drawback of the approach you are suggesting. {6 points}

Looking for :

Representation scheme for recipe

Discussion of whether the scheme is OOD

At least one drawback

3.d. Define a class to start testing the Dish class and the ingredient class. You should define the class and write a short main method that creates one ingredient object and one Dish object and fills them with data for your favorite ingredient and your favorite Dish. Make sure that your Dish object is connected appropriately to your raw ingredient in your data. {6 points}

public class DishTest

{

public void main ()

{

Ingredient greenGrapes = new Ingredient("green grapes", "fruit salad", KILOGRAMS, FRUITS, .33)

RecipeIngredient grapes = new RecipeIngredient(greenGrapes, .5);

// assume other ingredients like strawberries, melon, dressing ingredients…

RecipeIngredient[] fruitSaladFruits = {grapes, melon, honey, orangeJuice};

String[] fruitSaladRecipe = {"Get fruit into bite size pieces", "Mix honey and orangeJuice in small bowl", "Combine all fruit and mix gently", "Drizzle honey-OJ dressing over fruit mixture"};

Dish fruitSalad = new Dish("Fruit Salad", fruitSaladFruits, fruitSaladRecipe);

}

}

// green grapes $ 1.49 per pound

//1.49 per .453592 kilograms

//0.3284 per kilogram

3.e. For the same test class, how would you choose to store a set of recipes and a group of ingredients? Write no more than three sentences to define your choice for storing recipes and ingredients and explain why that approach was chosen. {6 points}

Three 3 sentences

Discuss recipe storage

Discuss ingredient storage

Explain why

3.f. Is object-oriented design a good fit for designing a recipe catalog? Why or why not?

{4 points}

Well-reasoned and specific answers are required

3.g. Given a set of recipes and a group of ingredients,, the "recipe catalog", what are at least three tasks that could be done using this information? Assume that you have some ingredients in your kitchen and you are looking at your recipe catalog. ( i.e. If you had the recipes and ingredients information, what could you do? This is similar to thinking about what things should you "test" in your main method.) List and describe the three tasks and how you would test that task. {9 points}

Things you could do with a recipe catalog: (must have 3 tasks and tests) The three below are EXAMPLES

Task: Determine recipes that could be made with current available ingredients. If you have a list of ingredients in the kitchen, you could search for recipes where all the recipe ingredients are available in the kitchen. This would require multiple cycles of matching.

Test with: One recipe which should have all ingredients available and one recipe which should not.

Task: Select dishes for a meal given certain constraints. Search for a dish the meets certain constraints like "does not include onions", "main ingredient is salmon", "is gluten-free". Would need to match ingredients or insure that an ingredient did NOT match.

Test with: Recipes that contain a variety of constraints on ingredients or on steps.

Task: Determine what additional ingredients are needed for a particular dish given the current ingredients. Similar to the earlier task but given a specific recipe, match as many ingredients as possible and then determine what ingredients are not listed and how much is needed. If an ingredient is partially matched (some amount but not enough), determine the needed amount. Make a list of the needed ingredients.

Test with: One recipe which should have all ingredients available; one recipe which has no ingredients available; one with full amounts of some ingredients and none of others; one with full amounts of some ingredients, partial amounts of others, and none of some ingredients.

4. Write the phrases defined by the acronyms below and give a quick description and definition of what each acronym is within Java. {3 points each; 9 total}

1. JVM Java Virtual Machine – the layer of the Java system that sits on top of the OS. The JVM is written to be native to the OS and it connects to the Java byte code. The Java byte code then can be identical across machines, allowing portability of Java code.
2. AWT Abstract Windowing Toolkit – "heavy" graphics components that connect to the specific machine for functionality. The Swing system sits on top of AWT and is platform-independent.
3. API Application Programming Interface – the library of written and tested Java classes that can be used by other developers

5. List three elements of Java syntax that are the same as C syntax and list three elements of Java syntax that are DIFFERENT from C syntax. Label the "same" syntax items. Give both the C version and the Java version for the "different" syntax items {6 points}

Must list three 3 SAME syntax elements and

must list three 3 DIFFERENT syntax elements

Must list both C and Java elements in each of the 6 examples

Must be syntax rather than other differences

6. Write the lines of Java needed to connect to the standard input stream for Java and then read in a string, then an integer, then a double. Make sure to declare all of the objects and variables needed to complete this task and make sure to write appropriate output statements to let the user know what information is requested. You may assume that the appropriate import statement is already written.

{6 points}

// Connect to the input stream

Scanner input = new Scanner(System.in);

String str; // be able to store the requested inputs

int I;

double d;

// Tell user what to do

System.out.printf("\nPlease enter a string: \n");

str = input.nextLine(); // read a string

System.out.printf("\nPlease enter an integer followed by a floating point number separated by a space. \n");

Ii= input.nextInt(); // read an int

d = input.nextDouble(); //read a double

Extra Credit questions

XC1. Give an example of a superclass that Dish (from 3.b.) might inherit from. Give a name and description of the superclass including names of at least two pieces of data and names of at least 2 methods that would be needed. The methods should NOT be accessors, mutators, or constructors. {6 points}

XC2. What is the ‘Garbage Collector’? {2 pts}

XC3. What do you like the best about your favorite dish? {Any meaningful answer will receive 2 points. ANY answer will receive at least 1 point.)