CSE1325 OOP Midterm

February 27, 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. Given the constructor below, declare the class that it is a constructor for, list all the data fields that the class would have, and list all the methods that it would need to have (you don't have to write them, just give the first line with access type, return type, name, and parameters). Basically, you are writing the class around the constructor. Assume the typical uses of private and public for access. [You may use the back of another page if needed but make a note on this page if the rest of the answer is somewhere else.] (16 points)

// class name and data fields up here

class ImFacultyMember extends ImUniversityEmployee

{

private boolean tenureTrack;

private AcademicTitle facultyTitle;

private int currentlyTeaching;

private int payCycle;

private double monthlyPay;

private double totalPay;

public ImFacultyMember(String lastName, String firstName, Dept whichDept, int yearStarted, boolean facultyStaff, boolean tenureTrack, AcademicTitle facultyTitle, int currentlyTeaching, int payCycle, double monthlyPay)

{

super(lastName, firstName, whichDept, yearStarted, facultyStaff);

//could any values in the line above be invalid?

this.tenureTrack = tenureTrack;

this.facultyTitle = facultyTitle;

if ( ! (setCurrentlyTeaching(currentlyTeaching))) // how many classes?

{

setCurrentlyTeaching(0);

}

setTotalPay( payCycle, monthlyPay);

}

// signatures of all needed methods in this class listed here

public void setTenureTrack(boolean tt);

public void setFacultyTitle(AcademicTitle ft);

public boolean setCurrentlyTeaching(int ct);

public boolean setPayCycle(int pc);

public boolean setMonthlyPay(double mp);

public boolean setTotalPay(int pc, double mp);

public boolean getTenureTrack();

public AcademicTitle getFacultyTitle();

public Int getCurrentlyTeaching();

public Int getPayCycle();

public double getMonthlyPay();

public double getTotalPay();

}

1.b What is the name of the enumerated type used in the super class ("parent class") of the faculty member class in question 1.a? Define the enumerated type and make sure it contains at least five unique values that correspond to UTA definitions. {6 pts}

enum Dept

{ MAE, BE, CE, CSE, EE, IE, MSE }

1.c For the data member called payCycle, the following are valid values : 12, 9, 4, and 1. If payCycle has one of these values, then a private data member called totalPay will be set to the value of monthlyPay times payCycle. If any other value is in payCycle, then totalPay is set to montlyPay.

Write the method setTotalPay (as it is used in part 1.a) to calculate the totalPay for a faculty member object and write the method setPayCycle that checks the validity of the payCycle value passed in and is called by setTotalPay. {12 pts}

public boolean setTotalPay(int pc, double mp)

{

if (setPayCycle( getPayCycle() ))

{ totalPay = getMonthlyPay() \* getPayCycle();

}

else

{ totalPay = getMonthlyPay();

}

}

public boolean setPayCycle(int pc)

{

boolean rPC = false;

switch(pc)

{ case 1:

case 4:

case 9:

case 12: rPC = true;

}

if (( pc < 0) || (pc > 12))

{ payCycle = 0;

}

else

{ payCycle = pc;

}

}

1.d Write a toString function for the class defined in 1a. If you refer to another class or toString method, you must also define that class or method. {8 pts}

Assume that the enum discussed in 1.b. and that the following enum are defined for you:

public enum AcademicTitle

{ PROFESSOR, ASSOCIATE\_PROFESSOR, ASSISTANT\_PROFESSOR, LECTURER, SENIOR\_LECTURER, INSTRUCTOR }

public String toString() // in ImFacultyMember, super class is ImUniversityEmployee

{

return String.format("%s is a %s %s currently teaching %d classes. They are on a %d pay cycle with a total pay of %f", super.toString(), (getTenureTrack() ? "tenure-track":"non-tenured"), getFacultyTitle(), getCurrentlyTeaching(), getPayCycle(), getTotalPay());

}

public String toString() // in ImUniversityEmployee

{

return String.format("%s %s, who works as a %s member in the %s department and started in %d, ", getFirstName(), getLastName(), (getFacultyStaff() ? "faculty":"staff"), getDept(), getYearStarted());

}

1.e. Give a short answer to explain why the constructor in 1.a assigns values directly for facultyTitle and tenureTrack but calls setCurrentlyTeaching and setPayCycle methods for those two values.

{8 pts}

The constructor sets facultyTitle and tenureTrack values directly because there is no error checking required for those values. The title string is not checked and the TT value is a boolean that can only have two values. The values for sCT and sPC are both numeric values. The values of these should be error checked therefore the set functions are called because they will verify the values.

1.f. If the class in question 1.a has a private data field called totalPay and its parent class also has a private data field called totalPay and both have get and set functions, how can the subclass setTotalPay method access the amount of totalPay from the super class? Write a Java method for the subclass setTotalPay function that takes no inputs but calculates totalPay for the ImFacultyMember class as the payCycle times monthlyPay (as described in 1.c.) PLUS the totalPay value that is stored in the superclass. {7 pts; up to 3 points extra credit for reusing your earlier code}

public boolean setTotalPay()

{

boolean wasPaySet = setTotalPay(getPayCycle(), getMonthlyPay());

if (wasPaySet)

{

totalPay = totalPay + super.getTotalPay();

}

return wasPaySet;

}

1.g. Given the constructor in question 1.a, write an object declaration (such as in a test class) that would declare an object of the ImFacultyMember class and would initialize all the values listed in the constructor. The new faculty member object should start in the year you were born and their facultyTitle should be ASSOCIATE\_PROFESSOR. The signature of the constructor is given below to remind you. {10 pts}

public ImFacultyMember(String lastName, String firstName, Dept whichDept, int yearStarted, boolean facultyStaff, boolean tenureTrack, AcademicTitle facultyTitle, int currentlyTeaching, int payCycle, double monthlyPay)

private ImFacultyMember drQ = new ImFacultyMember("Smythe", "Obinna", CSE,

\_\_\_\_\_year of your birth\_\_\_\_\_\_\_, true, true, ASSOCIATE\_PROFESSOR, 3, 9, 12000);

1.h. Write another object declaration (such as in a test class) that would declare an object of the ImFacultyMember class and would initialize all the values listed in the constructor. Include at least TWO values which could be error checked in set functions and found to be invalid. The signature of the constructor is shown in the previous question. {12 pts}

private ImFacultyMember drQ = new ImFacultyMember("Smythe", "Obinna", CSE,

-1942 , true, true, ASSOCIATE\_PROFESSOR, -7, -9, -1000);

The enumerated types are not actually error checked because it won't compile if they aren't valid.

2. Match the definitions below to the word or phrase they describe. {3 pts each; 21 total}

G Data abstraction F Encapsulation A Inheritance C Polymorphism

B Instantiation E Object-oriented D Signature

A. A way of being able to use public data members and member functions from another class as if they were defined within the current class; creating an *is-a* relationship between classes

B. The creation of a space in memory with an attached identifier to hold a all the elements of a class

C. The reuse of identifiers with unique signatures such that the user does not have to distinguish between different input types when applying the same process to the data

D. The combination of a method name, its return type, its access specifier, and its parameter list

E. An approach to programming the focuses on the data to be represented rather than the actions to be performed

F. Creating a class that contains all data and methods necessary to provide functionality but no more or less than that

G. Creating a class structure that can be used without knowledge of lower level implementation details

Extra Credit questions

XC1. What built-in Java class have we used to read input from the keyboard (without a graphical interface)? {2 pts; 2 extra points if you list and describe a graphical interface class also}

Scanner

JPanel – creates a window that can be used for communication to the user

XC2. How could the algorithm in the method materialFinder be simplified? {3 pts}

private Material materialFinder(String im) {

int ic =0;

if (im.equalsIgnoreCase("P")) { ic = 1; }

else if (im.equalsIgnoreCase("C") { ic = 2; }

else if (im.equalsIgnoreCase("O")) { ic = 3; }

else if (im.equalsIgnoreCase("G")) { ic = 4; }

switch (ic) {

case 1: return Material.PAPER;

case 2: return Material.CARDBOARD;

case 3: return Material.POLY;

case 4: return Material.GLOSSY\_STOCK;

default: return Material.MISC;

}

}

Remove the double layer of selection (if and then a switch based on the if results)

if (im.equalsIgnoreCase("P")) {return Material.PAPER;}

else if (im.equalsIgnoreCase("C") {return Material.CARDBOARD;}

else if (im.equalsIgnoreCase("O")) {return Material.POLY;}

else if (im.equalsIgnoreCase("G")) {return Material.GLOSSY\_STOCK;}

else {return Material.MISC;}

XC2. What is the ‘Garbage Collector’? {3 pts}

The Garbage Collector is the subsystem in the Java Environment that keeps track of used and available memory and works in the background to release memory that is not longer accessible to the program so that it can be reused.

XC4. What question would you have asked on this midterm? {Any meaningful answer will receive 2 points. ANY answer will receive at least 1 point.