CSE1325 OOP Test 1

March 7, 2013

Name: Key UTA ID: 1000

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. The test is worth 100 points and there will be 10 points extra credit available.
3. If you give an answer for the third extra credit question and then also give the real first and last name, you will get two additional points.
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.

1.a. Given a constructor as follows, describe (write down) everything (data, methods, etc.) you can determine about the associated class. (8 points)

public ThisAndThat( int height, int weight, Colors eyeColor, String hairColor, boolean organDonor)

{ super(height, weight);

 eyeC = eyeColor;

 hairC = new String(hairColor);

 organD = organDonor;

}

class name is ThisAndThat

class has at least five data fields:

integers height and weight // variables of type integer

Colors eyeC // object of class Colors

String hairC // object of class String

Boolean organD // a true / false

The class is an extended class from something else and it inherits height and weight from the superclass. The superclass constructor will set height and weight.

1.b. Assume that the code fragment (switch statement) below is valid code with no syntax errors. Does this code add information to your answer above? If yes, what info is added? If no, how is this code already accounted for? (8 points)

 switch (eyeColor)

 { case BROWN: System.out.println("Mmm, chocolate");

 break;

 case BLUE: System.out.println("Like the sky");

 break;

 case GREEN: System.out.println("Spring!");

 break;

 case RED: System.out.println("Vampire! Run away!");

 break;

 default: System.out.println("What color is that?");

 break;

 }

Yes, this gives enumerated type values for Colors. Colors is an enum not a class.

enum Colors {BROWN, BLUE, GREEN, RED}; //might be more colors

|  |  |
| --- | --- |
| 1 | import java.io.\*; |
| 2 | import java.util.\*; |
| 3 | public class WhatDoIDo { |
| 4 |  |
| 5 |  public static void main (String arg[]) { |
| 6 |  |
| 7 |  File file = null; |
| 8 |  if (arg.length > 0) { |
| 9 |  file = new File (arg[0]); |
| 10 |  } |
| 11 |  if (file == null ) { |
| 12 |  file = new File ("textOutput.txt"); |
| 13 |  } |
| 14 |   |
| 15 |  Scanner scanner = null; |
| 16 |  try { |
| 17 |  scanner = new Scanner(file); |
| 18 |  }  |
| 19 |  catch (FileNotFoundException e) { |
| 20 |  System.out.println ("File not found!"); |
| 21 |  } |
| 22 |  finally { |
| 23 |  System.out.println ("Ready"); |
| 24 |  } |
| 25 |   |
| 26 |  try { |
| 27 |  while (!(scanner.hasNextInt())) { |
| 28 |  scanner.next (); |
| 29 |  } |
| 30 |  while (scanner.hasNext()) { |
| 31 |  System.out.printf ("%3d \\" ,scanner.nextInt ()); |
| 32 |  System.out.printf ("%3d \\" ,scanner.nextInt ()); |
| 33 |  System.out.printf ("%5d \n" ,scanner.nextInt ()); |
| 34 |  System.out.printf ("$%6.2f \n",scanner.nextFloat ()); |
| 35 |  } |
| 36 |  } |
| 37 |  catch (InputMismatchException e) { |
| 38 |  System.out.println ("Mismatch exception:" + e ); |
| 39 |  } |
| 40 |  } // main |
| 41 |  |
| 42 | } // class  |

2. Use the code above to answer the questions that follow:

2.a. What is the name of the class? WhatDoIDo (2 pts)

2.b. List the data fields and methods of the class: (3 pts)

No data fields, only a main method

2.c. Explain how the user would give input for the class and how the class would find that input. I.e. what does the user have to do and then what does the program do with the user's action? (Also include in your explanation a description of what line 8 does related to this.) (6 pts)

The user passes in a file name when the class is invoked. The class finds the file by looking at the argument list from the arg parameter. In line 8 there is error checking to make sure that at least one argument is passed in to the program. If there is a file name, then the program reads from the file. If there is no file name, then the program uses a file with the default name textOutput.txt .

2.d. What is happening on line 17? Do not just turn the code into words – explain the action in words. (5 pts)

line 17 is setting up the program to be able to read from the file that is given. Java knows how to read from Scanner objects so a new Scanner object is initialized with the file as its value.

2.e. What exception handling is done in the program? Name the exception(s) and describe what is done with the exception(s) if caught. (6 pts)

The program checks for an exception in the expected data in the file, i.e. an inputMismatchException. This error means that one data type was expected but a different type was found. If this exception happens, then the program prints a message indicating the exception occurred.

2.f. If the file textOutput.txt contains the following data, what are the two lines of output produced by the program after the word "Ready"? (6 pts)

textOutput.txt

3

6

2013

43.78

7 4 1962

100.00

2 25 2005 1.08

The program produces :

3 / 6 / 2013

$ 43.78

2.g. What do the lines 27-29 accomplish? (5 pts)

The while loop goes through the file reading it and ignoring the data until it finds an integer

2.h. What do you think is represented by the data printed on lines 31-33? Explain why. (4 pts)

A date because there are three parts to it, separated by slashes

2.i. Overall, in three sentences or less, describe what the program does. (6 pts)

The program reads dates and money amounts from a file until the end of the file

3. Define the following Java vocabulary in your own words: (9 pts)

Inheritance: a way to establish Is-a relationship between objects

Instantiate: the creation of a real instance or particular realization of an abstraction or template such as a class of objects or a computer process. To instantiate is to create such an instance by, for example, defining one particular variation of object within a class, giving it a name, and locating it in some physical place.

Interface: a description of all functions that an object must have in order to be an "X"

4.a. Define part of a Java class that contains the class header and the data fields to represent the following information: the make, the model, the year, the current value, and the color of your dream car. Assume that the car colors are the same as those used in question 1. You may reuse your definition from question 1 without recopying. Follow the standard conventions for representing your data. {8 points}

class DreamCar

{

private String make;

private String model;

private int year;

private double value; //money amount needs float

private Colors color; //from ques 1

}

4.b. Write methods for your dream car class to allow the user to enter values for each of the data fields. The methods should error check the values as appropriate. If you error check the makes and models, do not check more than four specific values, e.g. you could check just Jeep, Mazda, BMW, Mini, and other. (If you need more space, use the back of the previous page.) {8 points}

// data fields - String make; String model; int year; double value; ThisAndThat.Colors color;

// Set methods

public boolean setYear(int yr)

{

if ((yr > 1905) && (yr <2015))

{

 year= yr; // no cars before 1905

 return true;

}

else

{

 return false;

}

}

public boolean setValue(int vl)

{

if (vl > 0) // value should not be negative

{

 value= vl;

 return true;

}

else

{

 return false;

}

}

public boolean setMake(String mk)

{

make= mk; // could do error checking but…

return true;

}

public boolean setModel(String md)

{

model= md; // could do error checking but…

return true;

}

// must either declare enum public or

// must use different type to pass in

public boolean setColor(char cl)

 {

 switch (cl)

 {

 case 'B':

 color = ThisAndThat.Colors.BROWN;

 break;

 case 'L':

 color = ThisAndThat.Colors.BLUE;

 break;

 case 'G':

 color = ThisAndThat.Colors.GREEN;

 break;

 case 'R':

 color = ThisAndThat.Colors.RED;

 break;

 default:

 color = ThisAndThat.Colors.BROWN;

 break;

 } // could do error checking but…

 return true;

 }

4.c. Write a constructor for the dream car class you created. Allow the user to enter values for the data in the new object. {8 points}

 public DreamCar(String mk, String md, int yr, double vl, char cl)

 {

 make = mk;

 model = md;

 switch (cl)

 { case 'B':

 color = ThisAndThat.Colors.BROWN;

 break;

 case 'L':

 color = ThisAndThat.Colors.BLUE;

 break;

 case 'G':

 color = ThisAndThat.Colors.GREEN;

 break;

 case 'R':

 color = ThisAndThat.Colors.RED;

 break;

 default:

 color = ThisAndThat.Colors.BROWN;

 break;

 }

 if ((yr < 1905) || (yr > 2014))

 { year = 2013;

 }

 else

 { year = yr;

 }

 if (vl < 0)

 { value = 0;

 }

 else

 { value = vl;

 }

 }

4.d. Define a class SportsCars that contains a collection of dream cars. Define the data field to hold the car collection info. Write one method that iterates through the collection to print all the models and the value of each model (Assume that a toString function is NOT available for the dream car class.) {8 pts}

public class SportsCars

{

 private DreamCar[] myCars;

 // assume proper constructor, set and get methods are defined

 public void printModnValue()

 {

 for (DreamCar car: myCars)

 System.out.println("Model is "+car.getModel()+" and value is "+car.getValue());

 }

}

5. List at least two advantages to using classes and objects and describe how Java provides those advantages. {8 points}

Many answers accepted

Extra Credit questions

XC1. Is there a parent object to the class used in question 2? If so, what class? {3 pts}

Yes, Object

XC2. If you were to implement exception handling on the code below, indicate where a try block would go and write the exception handler for the error you are considering. {5 pts}

public class EXC2 {

 public static int quotient(int numerator, int denominator)

 {

try

{

 return numerator / denominator;

}

catch(ArithmeticException e)

{

 String.format("Invalid division attmepted");

}

 }

 public static void main(String[] args) {

 int x = 0;

 int y = 4;

 quotient(y, x);

 }

}

XC3. What is the TA's favorite nickname for himself? {Any answer that I could share with my grandmother will receive 2 points. )

Sean Fierce (2 pts)

Sean Pierce (+2pts)