Paxton Proctor

CMPS 4143-101 Cont: Java Python

Prof Das

10/11/2021

Program Assignment #1

Problem #1

* Write an OOP code (A case study: Should be unique; If you are copying code from online you will be caught) that has the following OOP features: inheritance (any), polymorphism (runtime and compile time), abstraction and encapsulation

1. Output:

* Use of Inheritance, Polymorphism runtime and compile time, some abstraction and encapsulation.

1. Input:

* Integers, doubles, and strings

1. Algorithm:

* I created a couple classes for Inheritance and then I used an override and overloaded method. Then I created a couple of abstraction and encapsulation classes.

1. Steps:

* Created two classes one inherits from the other.
* Created two types of Polymorphism Override and overload
* Created two more Classes that used encapsulation and abstraction

Problem #2

* Consider the Following organigram(can be found in https://en.wikipedia.org/wiki/Mammal)of Mammalia family.(a: 5, b: 5, c: 10, d: 20, e:10)-(50 points)a. Each of members i.has 1.Legs (count)2. Tails (yes/no) 3.Brain (yes/no)4.Neurons (count) 5.Precision to understand (0.00 to 100.00%) d6. Birth date 7. Favorite foods ii. Can 1. Eat 2. Walk 3. Sleep 4. Give Birth b. Think about perissodactyla, they can mate with Carnivora members; can breed and produce a new family called ‘Perivora’. Perivora animals supposed to give birth to a new family ‘Periveron’. But by nature law no periveron member can be crated. c. Create at least 5 animals from each family; Perviveron member cannot be produced. d.Now you can do some creativity by changing the neurons, precision to understand, food habits, but you can’t change the legs, brains, birth date. Change these features for at least 20 animals and show how it was before and how it is now. From level 6 to 8(Considering Mammlia as level-1 and carnivora, cetartiodactyla are on level-8), animals can run and hunt for their food. Change at least 10 animals’ running speed, hunting process and hunting food type for those who are 6-8 level animals. Try changing at least 10 animals and Show (reading from files; see next bullet)what they were and what they are now. e. All outcomes should be stored in a file. When you have created objects, their information should be stored in files. After changing their behaviors and attributes, they should also be stored in files. The term Show mentioned in the previous subsection refers to reading the data from files and printing to the console

1. Output:

* Print At least 5 animals and their attributes, 20 animals with different attributes, change their speed and the way they hunt. And print it into an out file.

1. Input:

* Integers, doubles, strings, anything and everything

1. Algorithm:

* Use muli-level inheritance and good luck trying to make everything neat.

1. Steps:

* Make a base class and derive it 21 more times. Each class should have a super class that takes all the attributes and uses them. Multiple methods created so that you can edit and modify them to do what you want and last but not least print EVERYTHING.

Problem #3

Create a calculator class which can perform certain operations: addition, subtraction,

multiplication, division, and modulo. Now, add Java exception handling to your code by defining two exception classes SyntaxError and RuntimeError. A SyntaxError exception should be thrown when an illegal character is found, a closing) is not found, or a = is not used twice in an expression or an unwanted alphanumeric character inside the equation. A RuntimeError exception should be thrown when a divide by zero occurs. The exceptions should propagate the error to the main program which prints the diagnostics of the error.You must handle these errors using Java exceptions and the message should be printed by a Java exception handler in a catch clause.(30 points)

1. Output: Compile, run, and test your program with at least these expressions:
2. Test Expression
3. Correct Response
4. ----------------
5. ---------------------
6. X = 1+ 2 + (3
7. Syntax error: ‘)’ expected
8. Y = 2 + 5 = 3
9. Syntax error: unexpected ‘=’
10. Y =
11. 6\*Z+5
12. Syntax error:
13. more than one
14. variable
15. Y 3+5+(1+6)
16. Syntax error: ‘
17. =
18. ’ expected
19. Z = (3+5)/0
20. Runtime error: Divide by zero occurred
21. A = 7\*6/2
22. A = 21
23. Input:

* No Input

1. Algorithm:

* No algorithm needed

1. Steps:

* Make a runtime and a compile error class that throws a exception anytime it is called.