Rochester Institute of Technology
National Technical Institute for the Deaf
Information and Computing Studies Department

Name: Edward Riley

NACA.161 Programming Fundamentals II
In-Class Exercise #19 – Abstract Classes

Overview

This exercise is designed to review abstract classes. In particular, this exercise will require you to extend classes and differentiate between object types.

Note

Although it is BAD programming practice to have print statements inside constructors and mutators, you will be asked to do this in order to help you see what is going on inside the classes. Inheritance can be confusing and the print statements will hopefully help you understand the relationships between inherited classes.

MyClass Class

1) Create a class called **MyClass**. Save and compile the file.

2) Add the following private attributes:

Name	Type	Range of values
var1	int	10 to 20
var2	int	less than 0

3) Create an accessor for each attribute. Save and compile the file.

4) Create a mutator for each attribute.
   a) Make sure you store only the range of values listed in the table.
   b) If the parameter value is out of range, print an error message that states that an invalid value was entered. Make sure you include the name of the class, the name of the attribute and the value of the attribute.
   c) Save and compile the file.

What is the if-statement you used to test for a good value for var1?

if (_var1 < 10) else if (_var1 > 20)

What is the if-statement you used to test for a good value for var2?

if (_var2 <= 0)

5) Create a 2-argument constructor that
   a) Accepts 2 integers
   b) Uses the appropriate mutator to set the value of the attribute.
   c) Prints the message: "This is the superclass 2-argument constructor"

   Save and compile the file.

6) Create a default constructor that calls the 2-argument constructor with default values of 0 for both arguments. Also print a message that says:
   "This is the superclass default constructor"

   What keyword did you have to use to call the 2-argument constructor?

   MyClass();

   Save and compile the file.

Create a Test Class

7) Create a class called **Test** with a main method. Save and compile the file.

8) Create an object of type **MyClass** using the default constructor. Save and compile the file.

9) Compile and run the **Test** class.

   What runtime errors did you get?

   Input is out of range

10) Change the code in var2.

Does

10) Change the code in the default constructor to pass a value of 15 to var1 and –10 to var2.

Does it run without errors now? Yes

If you still have errors—fix them

Back to MyClass

11) Create a toString method in MyClass that returns the value of each attribute. Make sure you also identify the name of each attribute. Save and compile the file.

12) Create an abstract method called method1 in MyClass with no arguments and no return value.

What line of code did you enter for this method?

public abstract method1()

13) Compile the MyClass class (even if you know there is something wrong).

What compile error did you get?

MyClass is not abstract abstract methods can't have a body

invalid method declaration, return type required

Why did you get this error?

I received this error because myClass is not ~~abstract~~ + return as well as method body type requir

14) Fix the error and compile the MyClass class.

15) In Test, add a call to method1 using the MyClass object you created.

What compile error did you get?

myClass is abstract; cannot be instantiated

Why did you get this error?

MyClass is preventing me from accessing

Create a Subclass

16) Create a class called Subclass that inherits everything from MyClass. Don't add any code--just create the class header and a pair of curly braces.

What keyword did you use to inherit the class? extends

17) Save and compile the file.

What error did you get?

Subclass is not abstract and does not override method method1

Why did you get this error?

I need to define the remaing abstracts.

18) To fix this error you need to define the abstract method called method1 you inherited from the abstract class. Make this method print the string: "I am method1!". Save and compile the file. Don't continue until you get this class to successfully compile.

19) Create a default constructor that prints the message: "This is the subclass default constructor" and include your toString method to display your variables as well. Save and compile the file.

20) Create a 2-argument constructor that prints the message: "This is the subclass 2- argument constructor" and include your toString method to display your variables as well. Save and compile the file.

21) Add the toString method to each of your print statements in MyClass as well.

22) In the Test class, delete the line that attempted to create a MyClass object and add a line that creates an object of type Subclass using the default constructor. Compile and run.

Why does instantiating an object using Subclass instead of MyClass allow you to compile?

Because Subclass is not an abstract

Explain why each line displayed when you ran the Test class:

Line 1: "This is a superclass default constructor."
Subclass is calling Superclass

Line 2: "This is a subclass two arguments constructor."
Test is calling it with two arguments

Line 3: "ToString: var1 is var2 = 5"
I'm calling it inside obj method

Line 4: "ToString:
I'm calling it manually

23) Change the code in the Test class to call the 2-argument constructor for the Subclass object. (Use any 2 numbers for the arguments.) Compile and run.

Why did the superclass default constructor still run even though you ran the 2-argument subclass constructor?

Because Subclass is calling it.

24) Change the code in the Subclass 2-argument constructor to call the 2-argument constructor in its superclass with the values 10 and 20.

What does this code look like?

SubClass obj = new Subclass (10, 20)

25) Compile and run. Explain why each line displayed when you ran the Test class.

Line 1: "This is super default constructor"
class
SubString is calling it to being overridden

Line 2: "This is a subclass two argument constructor."
Test is Calling it.

Line 3:
To String is inside subclass method

When you complete all of the steps successfully and answer all of the questions, contact your instructor to check if your application(s) executes correctly and to review your code. We will initial the line below.

_____Successful execution of code

If you do not finish the program during the class period, contact your instructor to check to review your code and initial below.

_____Code not completed during lab time

You may then submit your work at the start of next class. You may not use the work period of the next class to complete this assignment. If you do not have a signature, then you cannot receive any points for this assignment.