Assignment 5

Arizona State University - CSE205- Assignment #5

Due Date Friday, February 7th, 5:30pm

Important: This is an individual assignment. Please do not collaborate.

No late assignment will be accepted.

Make sure that you write every line of your code. Using code written by someone else will be considered a violation of the academic integrity and will result in a report to the Dean's office.

All files must be submitted to Gradescope

Minimal Submitted Files

You are required, but not limited, to turn in the following source file:

Assignment5.java (https://canvas.asu.edu/courses/44324/files/12294899/download?wrap=1) (https://canvas.asu.edu/courses/44324/files/12294899/download?wrap=1) (Download this file and use it as your driver program for this assignment. You need to add more code to complete it.)

SummerCamp.java

DebateCamp.java

RoboticsCamp.java

MathCamp.java

SummerCampParser.java

Requirements to get full credits in Documentation

The assignment number, your name, StudentID, Lecture day/time, and a class description need to be included at the top of each file/class.

A description of each method is also needed. Some additional comments inside of methods (especially for a "main" method) to explain code that are hard to follow should be written.

New Skills to be Applied

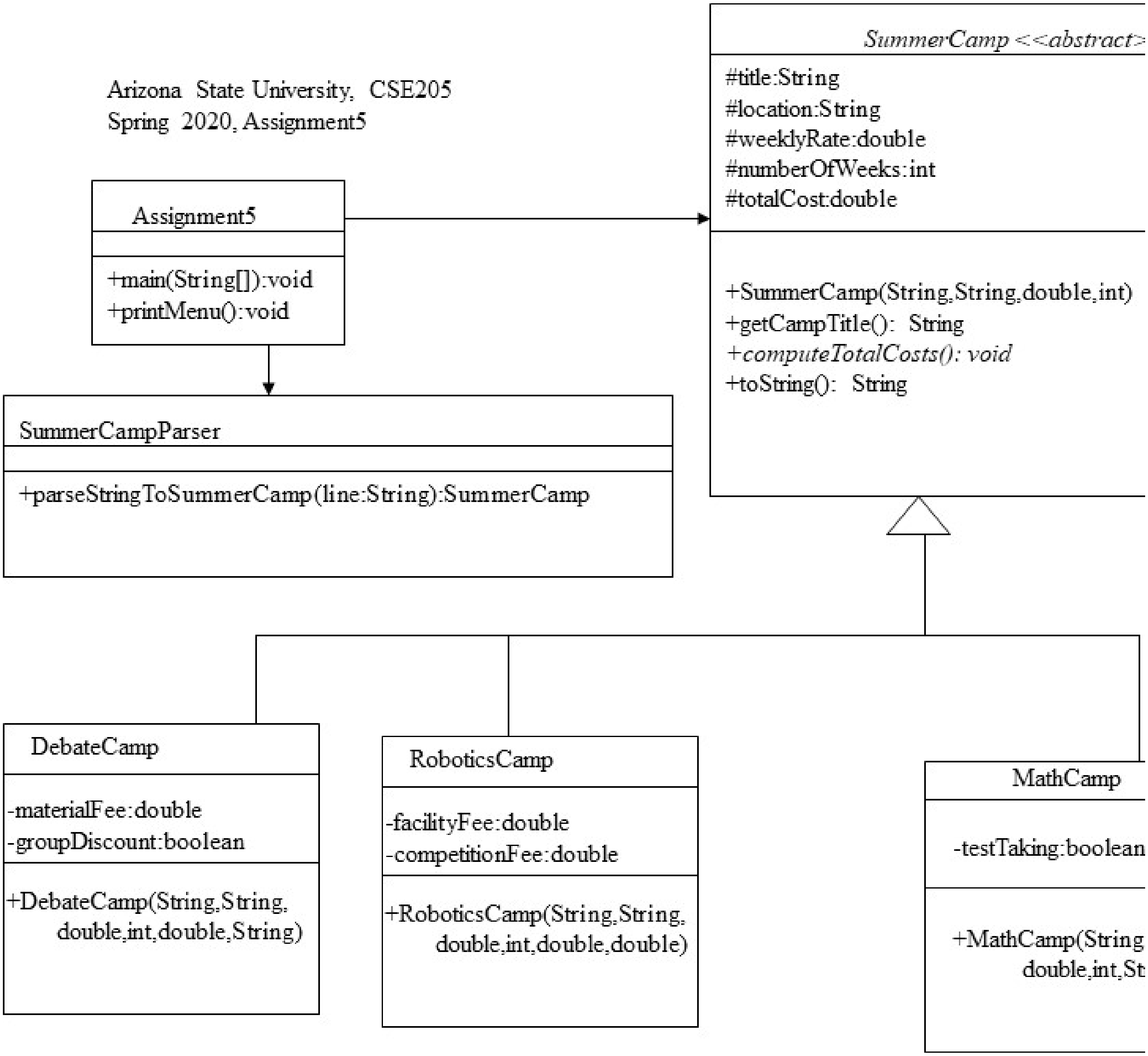
In addition to what has been covered in previous assignments, the use of the following items, discussed in class, will probably be needed:

- Inheritance
  The *protected* modifier

The *super* Reference
Abstract class
NumberFormat/DecimalFormat
ArrayList

Program Description

Class Diagram:



In Assignment #5, you will need to make use of inheritance by creating a class hierarchy for summer camps.

SummerCamp is an abstract class, which represents the basic attributes of any summer camp. It is used as the root of the summer camp hierarchy. It has the following attributes (should be protected):

Attribute name	**Attribute type**	**Description**
title	String	A title for a summer camp
location	String	A location for a summer camp
weeklyRate	double	some weekly Rate for a summer camp
numberOfWeeks	int	A number of weeks for a summer camp
totalCost	double	some total cost to be computed for a summer camp

The following constructor method should be provided to initialize the instance variables. Please refer to the UML class diagram for their parameter types, and return type.

***public SummerCamp(iString someTitle, String someLocation,
double someWeeklyRate, int someNumberOfWeeks)***

The instance variables title, location, weeklyRate, and numberOfWeeks are initialized to the value of the first parameter, the second parameter, the third parameter, and the forth parameter respectively. totalCost should be initialized to 0.0.

The following mutator method should be provided for instance variable the camp title

String getCampTitle( )

The SummerCamp also has an abstract methods (which should be implemented by its child classes:

void computeTotalCosts( )

The following toString method should be provided:

public String toString()

toString method returns a string of the following format:

\nCamp Title:\t\t*Robotics4\n*

Location:\t\t*Coor172\n*

Weekly Rate:\t\t*$70.00\n*

Weeks:\t\t\t3\n

Total Cost:\t\t$0.00\n

You should make use of the NumberFormat class (in java.text package) to format the weekly rate and total cost. NumberFormat should be used with an object of Locale (in java.util.Locale, i.e., Locale usMoney = new Locale("en", "US");) by specifying to use the US currency.

DebateCamp class

DebateCamp is a subclass of SummerCamp class. It has the following attribute in addition to the inherited ones:

Attribute name	**Attribute type**	**Description**
materialFee	double	some material fee for the debate camp
groupDiscount	boolean	where it has a group discount or not

The following constructor method should be provided:

public DebateCamp (String title, String location, double rate, int weeks, double materialFee, String discount)

The constructor of the parent class should be called using the first, second, third, forth, and fifth parameters to initialize its title, location, weeklyRate, numberOfWeeks, materialFee. If the parameter discount is yes, the instance variable groupDiscount should be set to true, and if it is no, groupDiscount should be set to false.

The following methods should be implemented:

public void computeTotalCosts( )

This method should update its totalCost as follows. Its basic total cost is computed as weeklyRate * numberOfWeeks. If groupDiscount is true, then it should get a 10% discount, i.e., weeklyRate * numberOfWeeks should be multiplied by (0.9). Then its materialFee should be added to it.

Also, the following method should be implemented:

public String toString()

The toString() method inherited from SummerCamp class should be used to create a new string, and display a savings account's information using the following format (note that if groupDiscount is true, it should show yes for Group Discount and it should show no otherwise):

\nDebate Camp:

\nCamp Title:\t\tDebateLevel5\n

Location:\t\tPSH153\n

Weekly Rate:\t\t$45.25\n

Weeks:\t\t\t4\n

Total Cost:\t\t$0.00\n

Material Fee:\t\t$12.00\n

Group Discount:\t\tyes\n\n

This toString method should make use of the toString method of its parent class.

You should make use of the NumberFormat class (in java.text package) to format the weekly rate, total cost, and material fee. NumberFormat should be used with an object of Locale (in java.util.Locale, i.e., Locale usMoney = new Locale("en", "US");) by specifying to use the US currency.

RoboticsCamp class

RoboticsCamp is a subclass of SummerCamp class. It has the following attribute in addition to the inherited ones:

Attribute name	**Attribute type**	**Description**
facilityFee	double	some facility fee for the robotics camp
competitionFee	double	some competition fee for the robotics camp

The following constructor method should be provided:

public RoboticsCamp(String title, String location, double rate, int weeks, double facilityFee, double compFee)

The constructor of the parent class should be called using the first, second, third, forth, fifth, and sixth parameters to initialize its title, location, weeklyRate, numberOfWeeks, facilityFee competitionFee.

The following methods should be implemented:

public void computeTotalCost( )

This method should update its totalCost as follows. Its basic total cost is computed as weeklyRate * numberOfWeeks. Then itsfacilityFee and competitionFee should be added to it.

Also, the following method should be implemented:

public String toString()

The toString() method inherited from SummerCamp class should be used to create a new string, and display a savings account's information using the following format:

\nRobotics Camp:

\nCamp Title:\t\tRobotics4\n

Location:\t\tCoor172\n

Weekly Rate:\t\t$70.00\n

Weeks:\t\t\t3\n

Total Cost:\t\t$0.00\n

Facility Fee:\t\t$37.50\n

Competition Fee:\t$17.50\n\n

This toString method should make use of the toString method of the parent class.

You should make use of the NumberFormat class (in java.text package) to format the weekly rate, total cost, facility fee, and competition fee. NumberFormat should be used with an object of Locale (in java.util.Locale, i.e., Locale usMoney = new Locale("en", "US");) by specifying to use the US currency.

MathCamp class

MathCamp is a subclass of SummerCamp class. It has the following attribute in addition to the inherited ones:

Attribute name	Attribute type	Description
testTaking	boolean	whether taking a test or not

The following constructor method should be provided:

public MathCamp(String title, String location, double rate, int weeks, String testTaking)

The constructor of the parent class should be called using the first, second, third, and forth parameters to initialize its title, location, weeklyRate, numberOfWeeks. If the parameter testTaking is yes, the instance variable testTaking should be set to true, and if it is no, testTaking should be set to false.

The following methods should be implemented:

public void computeTotalCosts( )

This method should update its totalCost as follows. Its basic total cost is computed as weeklyRate * numberOfWeeks. If testTaking is true, $25 should be added to it.

Also, the following method should be implemented:

public String toString()

The toString() method inherited from SummerCamp class should be used to create a new string, and display a savings account's information using the following format (note that if testTaking is true, it should show yes for Test Taking Discount and it should show no otherwise.):

\nMath Camp:

\nCamp Title:\t\tAlgebra1\n

Location:\t\tCoor120\n

Weekly Rate:\t\t$65.75\n

Weeks:\t\t\t4\n

Total Cost:\t\t$0.00\n

Test Taking:\t\tyes\n\n

This toString method should make use of the toString method of the parent class.

You should make use of the NumberFormat class (in java.text package) to format the weekly rate and total cost. NumberFormat should be used with an object of Locale (in java.util.Locale, i.e.,
Locale usMoney = new Locale("en", "US");) by specifying to use the US currency.

SummerCampParser class

The SummerCampParser class is a utility class that will be used to create an object of a child class of SummerCamp class from a parsable string. The SummerCampParser class object will never be instantiated. It must have the following method:

public static SummerCamp parseStringToSummerCamp(String lineToParse)

The parseStringToSummerCamp method's argument will be a string in the following format:

For a debate camp with its type "DebateCamp"

type:title:location:weeklyRate:numberOfWeeks:materialFee:groupDiscount

For a robotics camp with its type "RoboticsCamp"

type:title:location:weeklyRate:numberOfWeeks:facilityFee:competitionFee

A real example of this string would be:

DebateCamp:Intro:WGHL101:30.50:6:9.50:no

OR

RoboticsCamp:Robotics4:Coor172:70.0:3:37.50:17.50

OR

MathCamp:Algebra1:COOR120:65.75:4:yes

This method will parse this string, pull out the information, create a new object of its corresponding child class of the SummerCamp class with attributes of the object, and return it to the calling method. The type will always be present and always be either DebateCamp, RoboticsCamp, or MathCamp. (It can be lower case or upper case) You may add other methods to the child classes in order to make your life easier.

Assignment 5 class

In this assignment, download **Assignment5.java (https://canvas.asu.edu/courses/44324/files/12294899/download?wrap=1) (https://canvas.asu.edu/courses/44324/files/12294899/download?wrap=1)** file by clicking the link, and use it for your assignment. **You need to add code to this file.** The parts you need to add are written in the Assignment5.java file, namely for the four cases "Add SummerCamp", "Compute Total Costs", "Search for SummerCamp", and "List SummerCamps".

All input and output should be handled here. The main method should start by displaying this updated menu in this exact format:

Choice\t\tAction\n
------\t\t------\n
A\t\tAdd SummerCamp\n
C\t\tCompute Total Costs\n
D\t\tSeatch for SummerCamp\n
L\t\tList SummerCamps\n
Q\t\tQuit\n
?\t\tDisplay Help\n\n

Next, the following prompt should be displayed:

What action would you like to perform?\n

Read in the user input and execute the appropriate command. After the execution of each command, redisplay the prompt. Commands should be accepted in both lowercase and uppercase.

Add SummerCamp

Your program should display the following prompt:

Please enter some summer camp information to add:\n

Read in the information and parse it using the summer camp parser.

Then add the new object (created by summer camp parser) to the summer camp list.

Compute Total Costs

Your program should compute total costs for all summer camps created so far by calling computeTotalCosts method for each of them in the summer camp list.

After computing total costs, display the following:

total costs computed\n

Search for SummerCamp

Your program should display the following prompt:

Please enter a summer camp title to search:\n

Read in the string and look up the summer camp list, if there exists a summer camp object with the same summer camp title, then display the following:

SummerCamp found\n

Otherwise, display this:

SummerCamp not found\n

List SummerCamps

List all summer camps in the summer camp list. Make use of toString method defined in the child classes of the SummerCamp class.

A real example is looked like this:

\nDebate Camp:

\nCamp Title:\t\tDebateLevel5\n

Location:\t\tPSH153\n

Weekly Rate:\t\t$45.25\n

Weeks:\t\t\t4\n

Total Cost:\t\t$0.00\n

Material Fee:\t\t$12.00\n

Group Discount:\t\tyes\n\n

\nMath Camp:

\nCamp Title:\t\tAlgebra1\n

Location:\t\tCoor120\n

Weekly Rate:\t\t$65.75\n

Weeks:\t\t\t4\n

Total Cost:\t\t$0.00\n

Test Taking:\t\tyes\n\n

If there is no summer camp in the summer camp list (the list is empty), then display following:

no summer camp\n

Quit

Your program should stop executing and output nothing.

Display Help

Your program should re display the "choice action" menu.

Invalid Command

If an invalid command is entered, display the following line:

Unknown action\n

Test Cases

Download the following input files, and the following output files, and save them in the same directory as **Assignment5.java (https://canvas.asu.edu/courses/44324/files/12294899/download?wrap=1) (https://canvas.asu.edu/courses/44324/files/12294899/download?wrap=1)** is located.

input1.txt (https://canvas.asu.edu/courses/44324/files/12294887/download?wrap=1) (https://canvas.asu.edu/courses/44324/files/12294887/download?wrap=1)

input2.txt (https://canvas.asu.edu/courses/44324/files/12294888/download?wrap=1) (https://canvas.asu.edu/courses/44324/files/12294888/download?wrap=1)

input3.txt (https://canvas.asu.edu/courses/44324/files/12294889/download?wrap=1) (https://canvas.asu.edu/courses/44324/files/12294889/download?wrap=1)

input4.txt (https://canvas.asu.edu/courses/44324/files/12294891/download?wrap=1) (https://canvas.asu.edu/courses/44324/files/12294891/download?wrap=1)

output1.txt (https://canvas.asu.edu/courses/44324/files/12294892/download?wrap=1) (https://canvas.asu.edu/courses/44324/files/12294892/download?wrap=1)

output2.txt (https://canvas.asu.edu/courses/44324/files/12294893/download?wrap=1) (https://canvas.asu.edu/courses/44324/files/12294893/download?wrap=1)

output3.txt (https://canvas.asu.edu/courses/44324/files/12294894/download?wrap=1) (https://canvas.asu.edu/courses/44324/files/12294894/download?wrap=1)

output4.txt (https://canvas.asu.edu/courses/44324/files/12294895/download?wrap=1) (https://canvas.asu.edu/courses/44324/files/12294895/download?wrap=1)

Error Handling

Your program should be robust enough to handle all test cases above.

What to turn in:

-Submit your **Assignment5.java (hw5/Assignment5.java)** , SummerCamp.java, DebateCamp.java, RoboticsCamp.java, MathCamp.java, SummerCampParser.java files

using Gradescope-> Assignment5 from canvas.asu.edu. Make sure that it is passing all test cases. Otherwise you will lose points for test cases (8pts). You can submit multiple times until the assignment deadline.

Grading Criteria:

____/ 5 Documentation (Each class file needs to have a header with your name, your information, and program description, each method needs its description and comments within your code)

____/ 1 Indentation and spacing (easy to read)

____/ 6 Required classes/methods and functionalities implemented

____/ 8 Produces correct results (test cases – auto graded)

Total points: 20

ASU disclaimer (https://urldefense.proofpoint.com/v2/url?u=http-3A__www.asu.edu_asuweb_disclaimer_&d=DwMGAg&c=l45AxH-kUV29SRQusp9vYR0n1GycN4_2jlnuKy6zbqQ&r=G4pk4AH0grx-bdOJg-VYjyrj13_G9rkmBOCyuJ0dDOA&m=7ajJFS1rAyresA0lmSKpZHOsdmQHPNC8wMxp5kiP00g&s=VOCArkzEukCuLkXcSz9THqlWXf3F0A81z0HVXvFfKzE&e=)

Copying any content of this page will be a violation of the copy right.