**COMP 2150 – Spring 2020**

**Homework 5: Abstract/Interface/Polymorphism (113 points)**

Number of People: Individual. Feel free to ask me for help, or visit the Computer Science Learning Center (<http://www.memphis.edu/cs/current_students/cslc.php>), or visit the Fri/Sat 9 – 10 am supplemental class @DH249

Due: (March 8) (SUNDAY) 11:59 pm

Submission: Zip all your **Java source files** (the one with .java extension) (you can zip the entire project folder if using an IDE) into a single zip file and upload it to the proper folder in the eLearn dropbox at <https://elearn.memphis.edu>.

Coding Style: (use the Format tool in Eclipse) . Use standard Java naming conventions for **variableAndMethodNames**, **ClassNames**, **CONSTANT\_NAMES**. You must have comments for all your codes. Comments should focus on what and why and NOT how.

Graders:

**MdMaminur Islam** **(**[mislam3@memphis.edu](mailto:mislam3@memphis.edu)**) (Section 002 )**

**Zinat Ara (zara) (**[zara@memphis.edu](mailto:zara@memphis.edu)**) (Section 001/350)**

**Utsav Shrestha (**[**shrstha3@memphis.edu**](mailto:shrstha3@memphis.edu) **) (Section 001/350)**

**Anup Shakya (**[**ashakya@memphis.edu**](mailto:ashakya@memphis.edu) **(Section 001/350)**

**Sabira Khanam Shorna (**[**shorna.sabira@gmail.com**](mailto:shorna.sabira@gmail.com)**) (Section 002)**

Questions about grading? Please contact your grader first.

Grader: Rong Qi, [rqi@memphis.edu](mailto:rqi@memphis.edu). Questions about grading? Please contact her first!

**Objectives: : Abstract/Interface/Polymorphism**

1. (10 points) This question is about working with abstract class in Java program. Create two classes (A and B) with A has an abstract method ( void call ( )) and B extends from A. Class A has a concrete method callA ( ) that displays “concrete method callA() in A”. B is a concrete class extended from A. Write a client program to display the following:

B’s implementation of A's call( ) .

concrete method callA( ) in A

1. (9 points) This question is about implementing interface methods in java programming.

Create an interface type A1 that has two methods (void m1 ( );, void m2 ( ); ).

Create another interface type B1 that extends interface A1 plus a method ( void m3( ); ) of its own.

Write a Class (MyClass ) to implement interface B1.

Each of the above methods (m1 ( ), m2 ( ), and m3 ( ) ) displays a message about their methods.

For example m1( ),

when invoked will display “ implement m1 ( ) “.

Write a client program to display the following 3 messages using one object.

Implement m1().

Implement m2().

Implement m3().

1. (14 points) Abstract Class Exercise. I used this program (NoGo.java) for one of my CS2 java lab exercises. There are number of steps that you need to follow along the way to collect points. The given NoGo class has two abstract classes (Nogo1, Nogo2) and an empty static main( ) method. NoGo.java has no compile error, and you can run NoGo.java without runtime error. It does not display any messages. Your task is to add codes along the following steps:
2. (2 points) create instance variables for the two classes (Nogo1, Nogo2) and explain the outcome.
3. (3 points) Create a subclass Go1 with Nogo1 as its parent and create a constructor that prints the “in Go1() constuctor)” message
4. (2 points) Create an instance of Go1 in main( ), run it, explain the outcome
5. (2 points) In the Nogo2 class, add an abstract method nMethod( );.
6. (3 points) Create a new class Go9 that extends the Nogo2 class;

Write a constructor for Go9 that displays (“in Go9() constructor)”)

write a method that overrides the method that you created in step 4. This new method displays “get nMethod( ) out of abstraction”.

1. (2 points) In the main( ) method , add some codes to invoke the method nMethod( );

Your final display should be:

NoGo1()

in Go1() constuctor

in Go9() constructor

got nmethod() out of abstraction

1. (15 points) Interface Exercise.

For this question, you need to create two interface programs: One of them has two methods ( countOne(), countTwo() ) and the other one has only one method (countThree ( )). In a separate package, write a class to implement the two interfaces. you need to import the interface program to a class where you implement the methods. You can create a new package in your assignment4 project (or package) to contain the two interface programs.

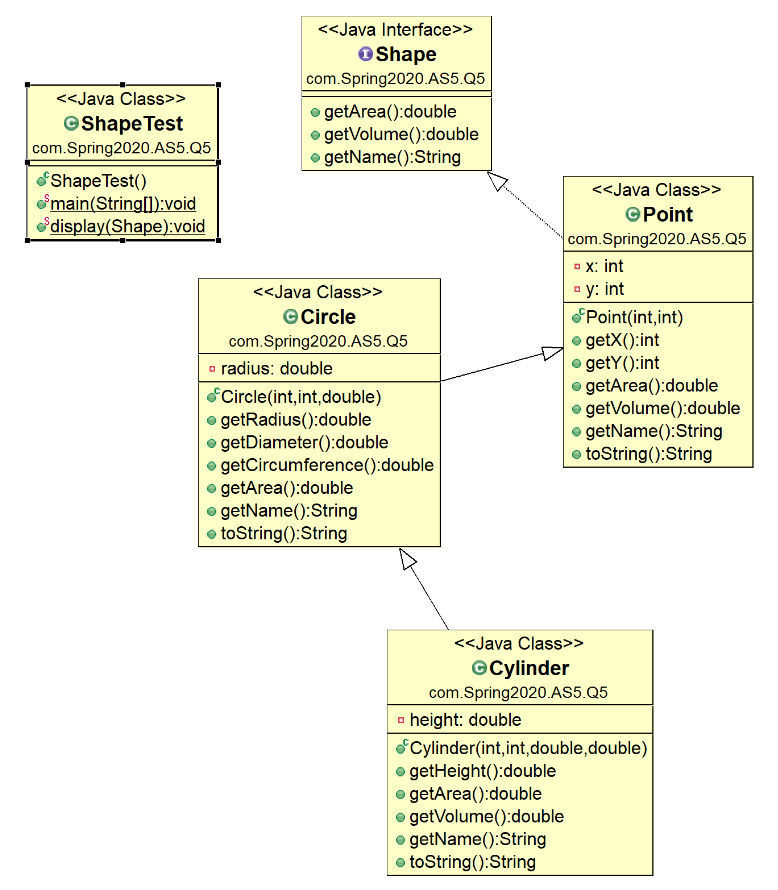
Create a client program to invoke the three methods to display the following:

ONE

TWO

THREE

1. (25 points) Create a program that calculates the areas and volumes of circle and cylinder using an interface class and inheritance. The UML diagram for the classes and the polymorphic interface for the Shape hierarchy class are shown:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | getArea() | getVolume() | getName() | toString() |
| Shape | 0.0 | 0.0 | abstract | Default Object Implementation |
| Point | 0.0 | 0.0 | “Point” | [x, y] |
| Circle | πr2 | 0.0 | “Circle” | center = [x, y] radius = r |
| **Cylinder** | 2πr2 + 2πrh | πr2h | “Cylinder” | center = [x, y] radius = r height = h |

Results Example:

Point location[5, 12]

Circle location[19, 9]; Radius = 5.5

Area = 95.03317777109123; Perimeter = 34.55751918948772

Cylinder location[20, 30]; Radius = 3.3

Area = 291.3198867673815; Perimeter = 20.734511513692635;

Height = 10.75

1. (25 points) For this exercise, you are given two interfaces to start building action characters needed to become Superman. Download the codes (Action.java) from eLearn and study the codes carefully. The program action.java runs without any changes. It should display the following:

I am Superman I can do the following:

fight bad guys

fly faster than Raptor F22

To be Superman, you need to add a few more actions using interfaces: (10 points)

Using the code from the following as reference

interface CanFight { void fight() ; }

Add an interface called Canfly

Add an interface called CanClimb

Add an interface called CanSwim

The CanSwim interface has two abstract methods:

void swimFast();

void treadH2O();

You need to modify the Action class to implement the new interfaces.

You also need to add “yourself” (your name) as another class type as that of Superman. ( 5 points)

You need to modify the Action class (the one with the main ( … ) ) to make your final result looks like the following:

I am SuperMan, I can do the following:

fight bad guys

Swim faster than Michael Phelps

Tread water forever

fly faster than Raptor F22

climb slippery cliffs

We all fight as an ActionChar

-------------

I am Mr.James Bond, I can do this:

Fight with smart gadgets

swim fast with scooter

Tread water 4ever with floats

fly fast with jet-vest

climb with robot

We all fight as an ActionChar

------ notice there are two fight() methods in the list. How do you separate them? Suppose you want the “two fights” associated to their own type, what do you do? (5 points)