

Object Oriented Programming 1

Spring 18-19

Lab Manual: 06

Lab Task:

1. Lab Review, and start with unfinished classes from Lab_05
2. Develop Java classes

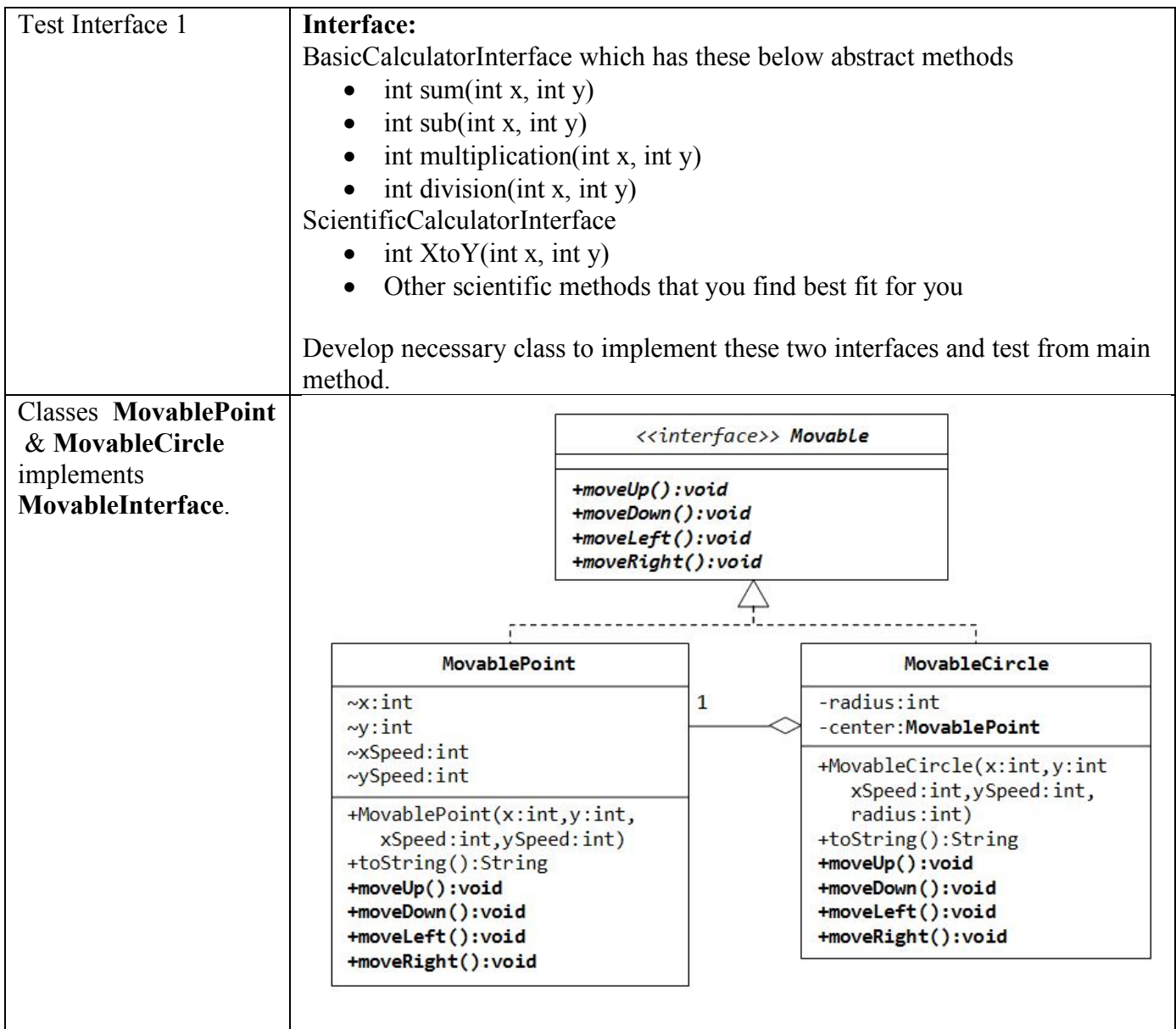
Note: Student must follow the name of class, member variables, and functions.

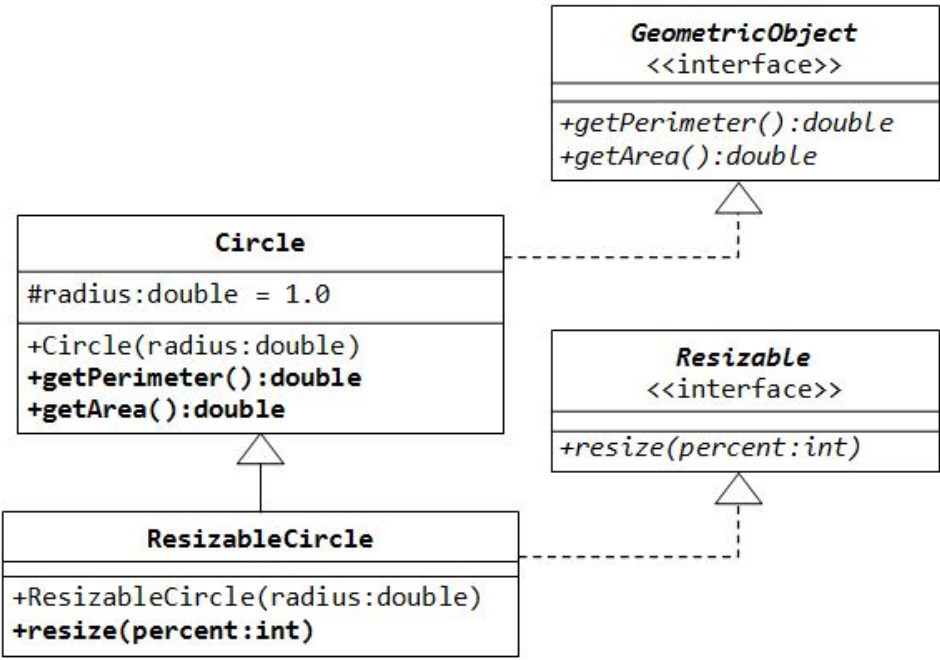
And students should use fully qualified names for these, as well camel notions.

And the syntax alignment has to be as it should be.

1. Lab_05 Classes – Review:

2. Develop Java Classes:



<p>Class Circle implements Interfaces GeometricObject</p> <p>Class ResizableCircle implemets ResizableInterface & inherits/extends</p>	 <pre> classDiagram class GeometricObject { <<interface>> +getPerimeter():double +getArea():double } class Resizable { <<interface>> +resize(percent:int) } class Circle { #radius:double = 1.0 +Circle(radius:double) +getPerimeter():double +getArea():double } class ResizableCircle { +ResizableCircle(radius:double) +resize(percent:int) } GeometricObject .. > Circle Resizable .. > ResizableCircle Circle .. > ResizableCircle </pre> <p>The diagram illustrates the relationships between four classes/interfaces:</p> <ul style="list-style-type: none"> GeometricObject (Interface): <ul style="list-style-type: none"> Methods: <code>+getPerimeter():double</code>, <code>+getArea():double</code> Resizable (Interface): <ul style="list-style-type: none"> Method: <code>+resize(percent:int)</code> Circle (Class): <ul style="list-style-type: none"> Attribute: <code>#radius:double = 1.0</code> Methods: <code>+Circle(radius:double)</code>, <code>+getPerimeter():double</code>, <code>+getArea():double</code> ResizableCircle (Class): <ul style="list-style-type: none"> Methods: <code>+ResizableCircle(radius:double)</code>, <code>+resize(percent:int)</code> <p>Relationships:</p> <ul style="list-style-type: none"> Circle implements GeometricObject (indicated by a solid line with an open triangle head). ResizableCircle implements Resizable (indicated by a solid line with an open triangle head). Circle inherits from ResizableCircle (indicated by a solid line with an open triangle head). Circle has a dashed association line to GeometricObject. ResizableCircle has a dashed association line to Resizable.
<p>Try this</p>	<p>Consider a bank has three different types of accounts. An account holder (Person/Student) can have any one of these below type.</p> <ol style="list-style-type: none"> 1. Current 2. Savings 3. Overdraft <p>These three types of account has different advantages and disadvantages</p> <ul style="list-style-type: none"> • Current Account <ul style="list-style-type: none"> ○ Current account holders can withdraw all of his/her amount • Saving Account <ul style="list-style-type: none"> ○ Saving account holders can withdraw maximum 80% of his/her total amount • Overdraft account <ul style="list-style-type: none"> ○ Overdraft account holders can withdraw additional amount which is set by at account creation time (overdraft limit amount) <p>After implement this structure change the account type (ex: current to saving or else) and changes will automatically reflected as well. You have to develop this scenario on your own.</p>

Special Note: **Do not** code interface and class on **same** .java file. Use **separate files** for interfaces. The interface should be **public**.