

Please read Sections 3.3 through 3.7 of the textbook and then answer the following, trying not to look at your notes or at the textbook. Quiz #4, on Fri. 16 Sep., will consist exclusively of questions taken from the Part 1 of this homework.

Part I — Questions

- Ex. 1. What is the difference between an actual argument and a formal parameter?
- Ex. 2. Write an accessor for a field of type `String` named `myName`.
- Ex. 3. Write a mutator for a field of type `int` named `myAge`.
- Ex. 4. What is the keyword used to tell the method to send a value back to the statement that called it?
- Ex. 5. How do we designate a method that returns the content of a field?

Ex. 6. In the following statement, what is the relationship between `myBox` and the method?

```
myBox.setLength(3.0);
```

- Ex. 7. Do different instances of the same class have access to the same methods?
- Ex. 8. Imagine that the following line appears in a UML diagram:

```
+ piesPerPerson(int people, double pies) : double
```

How many parameters has this method? Is it likely that their order matter?

- Ex. 9. What is the return type of a constructor?
- Ex. 10. How many arguments takes the default constructor?
- Ex. 11. What is the value assigned to a `bool` field by the default constructor?
- Ex. 12. Write a single statement which would do the same as the following, but uses the `new` keyword:

```
String name = "I'm an instance of the String class.";
```
- Ex. 13. Describe what an uninitialized variable is.
- Ex. 14. Is `myCustomConstructor` a valid name for a constructor in the class `CircleDemo`?
- Ex. 15. How do we call a constructor that does not take arguments?
- Ex. 16. Define what the signature of a method is.
- Ex. 17. When is a default constructor provided for a class?
- Ex. 18. When should you import an API package?
- Ex. 19. What determines the scope of a variable?
- Ex. 20. How do we call variables declared in a method body? In a method header?

Part II — Programming Exercises

Ex. 1. Write a `Circle` class with one field—a `double` named `radius`—and four methods:

1. a setter for `radius`,
2. a getter for `radius`,
3. a `getArea` method that returns the area of the circle ($\pi \times (\text{radius}^2)$),
4. a `getCircumference` method that returns the circumference of the circle ($2 \times \pi \times \text{radius}$).

Write also a constructor that takes one argument of type `double`, the radius of the circle created. Remember that you can access to an approximation of π using the constant `Math.PI` in your program, and that you can also access to the `Math.pow` method, which takes two `double` and return the first argument to the power of the second, without loading any package.

When you think you program is flawless, type it and compile it in BlueJ.

Ex. 2. Write an application program for the `Circle` class that you previously defined. This program should print the UML diagram for the `Circle` class, ask the user for a radius, create an instance of the `Circle` class with the radius given by the user, and print the radius, the area and the circumference of the circle created.

Using the `printf` method and some nice layout (including printing only two decimals) is strongly encouraged. When you think you program is flawless, type it and compile it in BlueJ.

