CS159 – Advanced Programming **James Madison University**
Sample Examination 1 **Fall 2023**

This work complies with the JMU Honor Code.
Name:_____ Signature:_____

Instructions. Answer all of the following questions. This is a "closed book" examination and you must work entirely on your own. You may not use a computing or communications device of any kind. Your answers need not conform to the course style guide. All questions that involve code or are about code use the Java programming language.

1. (10 points) Choose the best answer to each of the following:

(1) ____ In Java, `<` is

a. A relational operator
b. A unary operator
c. A distributed operator
d. All of the above
e. None of the above

(2) ____ In Java, `&&`

a. Is a logical operator
b. Is a binary operator
c. Has boolean operands
d. All of the above
e. None of the above

(3) ____ In Java, the `main()` method of the "main class" must have what return type?

a. `double`
b. `int`
c. `String`
d. `String[]`
e. `void`

(4) ____ In Java, the `main()` method of the "main class" must have a parameter of what type?

a. `double`
b. `int`
c. `String`
d. `String[]`
e. `void`

(5) ____ In Java, the parameter in the `main()` method of the "main class" is which of the following:

a. A primitive type
b. A reference type
c. A formal type
d. All of the above
e. None of the above

(6) ____ In the Java **statement** `i = (int)d;` the operator = is a/an:

a. Assignment operator
b. Relational operator
c. Typecast operator
d. All of the above
e. None of the above

(7) ____ In Java, given the following statements:

```
String[] months;
months = new String[12];
System.out.println(months[0]);
```

`months[0]` is which of the following?

a. A primitive type
b. A reference type
c. A formal type
d. All of the above
e. None of the above

(8) ____ In Java, what kind of **statement** is `int i;`?

a. Assignment statement
b. Declaration statement
c. Typecast statement
d. All of the above
e. None of the above

(9) ____ In Java, String variables should be compared using which of the following?

a. The `==` operator because `String` is a primitive type
b. The `equals()` method because `String` is a primitive type
c. The `equals()` method because `String` is a reference type
d. All of the above
e. None of the above

(10) ____ In Java, a variable is

a. The implicit loss of precision in arithmetic operations
b. The way the decrement operator changes
c. A named space for holding a value
d. All of the above
e. None of the above

2. (10 points) Consider the following implementation of the `Calculator` class.

```
public class Calculator {

    public static final int MAX_SIZE = 100;

    public static double circumference(double radius) {

        double result;

        result =  2.0 * Math.PI * radius;
        return result;
    }
}
```

Identify the best description of each of the following as it is used in this class diagram and fragment. You may use a description more than once.

_____	`Calculator`
_____	`circumference()`
_____	`Math`
_____	`Math.PI`
_____	`MAX_SIZE`
_____	`radius`
_____	`result`
_____	`2.0`
_____	`.`
_____	`*`

a. An actual parameter
b. An arithmetic operator
c. A "class constant"
d. The command-line arguments
e. A formal parameter
f. A literal
g. A local variable
h. A logical operator
i. The membership operator
j. The name of a class
k. The name of a method
l. A relational operator
m. A typecast operator

3. (6 points) Choose the best answer to each of the following:

(1) ____ In Java, an operator that has numeric operands and evaluates to a number is:

a. An arithmetic operator
b. A logical operator
c. A relational operator

(2) ____ In Java, an operator that has numeric operands and evaluates to a `boolean`:

a. An arithmetic operator
b. A logical operator
c. A relational operator

(3) ____ In Java, an operator that has `boolean` operands and evaluates to a `boolean` is:

a. An arithmetic operator
b. A logical operator
c. A relational operator

4. (4 points) Choose the best answer to each of the following:

(1) ___ In Java, the **statement** `doubl cost;`

a. A compile-time error
b. A run-time error
c. A repetitive stress error
d. All of the above
e. None of the above

(2) ___ In a Java program written for CS159, omitting a period from the end of the first line of a comment will cause which of the following?

a. A compile-time error
b. A run-time error
c. A Checkstyle error
d. All of the above
e. None of the above

(3) ___ In Java, assuming that the variables have been both declared and initialized, the **statement** `average = total / 0;` will cause which of the following?

a. A compile-time error
b. A run-time error
c. A repetitive stress error
d. All of the above
e. None of the above

(4) ___ In Java, assuming the following declaration and initialization:

```
String current;
String[] months;
months = new String[12];
```

the **statement** `current = months[12];` will cause which of the following?

a. A compile-time error
b. A run-time error
c. A repetitive stress error
d. All of the above
e. None of the above

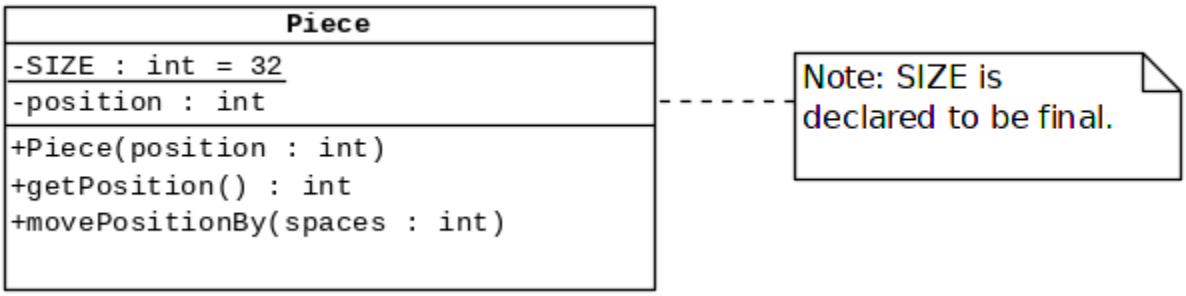
5. (5 points) Given the attached UML class diagram and specifications for the `Piece` class, implement the constructor. Your answer must include both the declaration and the body of the constructor.

6. (5 points) Given the attached UML class diagram for the `Piece` class, implement the accessor method. Your answer must include both the declaration and the body of the method.

7. (10 points) Given the attached UML class diagram and specifications for the `Piece` class, implement the mutator method. Your answer must include both the declaration and the body of the method.

Attachments

The `Piece` class is an encapsulation of a game piece in a board game.



1. The `position` Attribute
   1.1. Holds the value of the current position on the board.
      1.1.1. There are `SIZE` positions on the board (which are arranged in a circle, increasing in the clockwise direction).
      1.1.2. The position on the board must be 0-based.
2. The Constructor
   2.1. Must initialize the attribute named `position` to `0` whenever the parameter named `position` is invalid (whether negative or positive).
   2.2. Must initialize the attribute named `position` to the value of the parameter named `position` otherwise.
3. The `movePositionBy()` Method
   3.1. The parameter `spaces` contains the number of spaces that the piece must move (which can have any value) in the clockwise direction.
   3.2. This method must adjust the `position` attribute by the value of `spaces` (accounting for the number of positions and the fact that the board is circular). For example, if `position` is initially `18` and `spaces` is `22` it must assign `8` to `position`.