

## 1. Match the keyword to its definition. (12 points)

- |               |   |
|---------------|---|
| ___ void      | a. Used to refer to fields and methods of the current class   |
| ___ public    | b. Indicates that a method may cause a specific exception to occur                                      |
| ___ private   | c. Indicates that a method does not return anything   |
| ___ extends   | d. Indicates that a variable has a constant, never-changing value                                       |
| ___ final     | e. Used when creating an object to set aside memory   |
| ___ static    | f. Used to refer to fields and methods of the parent class  |
| ___ this      | g. Indicates that a method or variable can be accessed from anywhere                                    |
| ___ super     | h. Indicates an inheritance relationship between two classes  |
| ___ new       | i. Indicates the resultant value/variable of a method   |
| ___ return    | j. Indicates that the method or variable cannot be accessed outside the current class                   |
| ___ throws    | k. Indicates that the method or variable does not require an object to be created for it to be accessed |
| ___ try/catch | l. Used to handle exceptions  |

## 2. When does a constructor get called? (2 points)

- It is called every time an object is used to call a method
- It is called when an object is created
- It is a special type of method that never gets called and thus never runs
- It is called when the object is closed

## 3. What is the difference between a class and an object? (2 points)

- Each project can only have one object but multiple classes
- Java does not have objects, only classes
- A class is a physical entity, while an object is just a template
- An object is an instance (or version) of a class

4. Write one line of code to create an object for a class called GiftCard, utilizing a constructor that has four parameters: the first is a double for the amount of the card, the second is a boolean to indicate whether or not the card has been activated, the third is an integer for the pin, and the fourth is a String for the name of the store the gift card is for. (3 points)
- 

5. What is encapsulation? (2 points)

- a. The concept of keeping everything inside one class
- b. The concept of surrounding each method body in curly braces
- c. The concept of controlled access to prevent unchecked modification of class fields
- d. The concept of putting one loop inside another loop

6. What is polymorphism? (2 points)

- a. The idea that a child class can be referred to by the type of any of its parent classes
- b. The idea that one class can modify another class
- c. The idea that multiple classes can be declared in the same file
- d. The idea that one piece of code can be written many different ways

7. Write one benefit of using a relative path instead of an absolute path: (2 points)
- 
- 

8. What is a buffer in the context of File I/O? (2 points)

- a. The space between the start of the file and the first word in the file
- b. The Java class used to read information from a file
- c. The place in memory where the file contents are stored temporarily while reading/writing
- d. The reference to the location in memory where the file contents are stored temporarily

9. Label the following as either primitive types (P) or objects (O): (9 points)

___ Scanner	___ int[]	___ double
___ long	___ ArrayList	___ File
___ boolean	___ float	___ String

10. Fill in the print statement of the ArrayList after the following code executes: (4 points)

```
import java.util.*;

public class Testing {
    public static void main(String[] args) {
        ArrayList<String> gifts = new ArrayList<>();

        gifts.add("puzzle");
        gifts.add("shirt");
        gifts.add("socks");
        gifts.add("headphones");

        for (int i = 0; i < gifts.size(); i++) {
            if (gifts.get(i).charAt(0) == 'p') {
                gifts.set(i, "game");
            }
            else if (gifts.get(i).equals("socks")) {
                String temp = gifts.get(1);
                gifts.set(1, "socks");
                gifts.set(i, temp);
            }
        }
        System.out.println(gifts);
    }
}
```

[ \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_ ]

11. What gets printed out after the following code executes? (2 points)

```
public class Testing {
    public static void main(String[] args) {
        double[] heightsInches = {44.5, 74.25, 36.75, 55.5};
        System.out.println(heightsInches);
    }
}
```

- a. [44.5, 74.25, 36.75, 55.5]
- b. 44.5 74.25 36.75 55.5
- c. [D@7b23ec81
- d. NullPointerException

12. Write a loop that iterates through an array of integers. The loop should take two adjacent non-overlapping values and determine the remainder of dividing the first value by the second. The remainders should then be summed up, and the sum should be printed out after the loop completes. You can assume the array always has an even size. (6 points)

**Example:** If the array starts with: [12, 4, 13, 7, 24, 5]

- First, you should find the remainder of 12 divided by 4
  - o The remainder is 0.
- Then, you should add that to the remainder of 13 divided by 7
  - o The remainder is 6. 6 plus 0 is 6.
- Then, that total should be added to the remainder of 24 divided by 5
  - o The remainder is 4. 4 plus 6 is 10.
- The final result is 10

**Hint:** This can be done with only one line of code inside the loop. If you can't remember a certain method or operator, do your best to explain what you are thinking of.

```
public class Testing {  
    public static void main(String[] args) {  
        int[] nums = {12, 4, 13, 7, 24, 5};  
        int remainSum = 0;  
  
        for (                ) {  
  
        }  
        System.out.println(remainSum);  
    }  
}
```

13. Write the **header** for a method in the main class that takes in a *numerical* year, *text* month, and *numerical* day, and returns the total *decimal* snowfall for that day. Then, match the labels to each part of your method header. (10 points)

14. What is overloading? (2 points)

- a. Using too many parameters for a method
- b. Putting more than one class in the same file
- c. Rewriting a method from a parent class in a child class
- d. Creating two methods with the same name but different parameters

15. What is overriding? (2 points)

- a. Rewriting a method from a parent class in a child class
- b. Throwing an exception from a method
- c. Using try/catch to handle an exception
- d. Creating two methods with the same name but different parameters

16. Fill in the missing details (red arrows) in the code below. The class should follow proper encapsulation techniques for full credit. The constructor should initialize the fields to the parameter values. (8 points)

```
public class Holiday {  
    ----- String name;  
    ----- int day;  
    ----- String month;  
    public Holiday(String name, int day, String month) {  
        -----  
    }  
    public void setDay(int newDay) {  
        if (newDay > 0 && newDay < 32) {  
            -----  
        }  
    }  
    ----- () {  
        return day;  
    }  
}
```

17. The following questions are all based on this code snippet:

```
import java.util.*;

public class Testing {
    public static void main(String[] args) {
        ArrayList<Integer> numbers = new ArrayList<>();
        numbers.add(1);
        numbers.add(2);
        numbers.add(3);
        int favNum = 8;

        modify(numbers);
        modify(favNum);
        System.out.println(numbers);
        System.out.println(favNum);
    }

    public static void modify(ArrayList<Integer> nums) {
        for (int i = 0; i < nums.size(); i++) {
            nums.set(i, 0);
        }
    }

    public static void modify(int num) {
        num = 0;
    }
}
```

i. Fill in the output of the code: (2 points)

[ \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_ ]

\_\_\_\_\_

ii. Match the following: (2 points)

\_\_\_\_\_ demonstrates pass by value

a. The first method

b. The second method

\_\_\_\_\_ demonstrates pass by reference

iii. What is one reason why we don't we have to create an object to call modify()? (2 points)

\_\_\_\_\_

18. The following questions are all based on the following code snippet: (9 points)

```
import java.util.Arrays;

public class testing {
    public static void main(String[] args) {
        boolean[][] classSectionGrades = { {true, true, false, false},
                                             {false, false, false, false},
                                             {true, false, true, false} };

        boolean[] results = new boolean[3];
        int i = 0;
        for (boolean[] section : classSectionGrades) {
            boolean allFailed = true;
            A System.out.println(section);
            B System.out.println(Arrays.toString(section));
            for (boolean isStudentPassing : section) {
                C System.out.println(isStudentPassing);
                if (isStudentPassing) {
                    allFailed = false;
                }
            }
            results[i] = allFailed;
            i++;
        }
        D System.out.println(Arrays.toString(results));
    }
}
```

i. The first time the outer loop runs, what would A output?

---

ii. The first time the outer loop runs, what would B output?

---

iii. The first time the inner loop runs, what would C output?

---

iv. After the code finishes running, what would D output?

---

19. What is the output of the following code? (5 points)

```
public class Testing {
    public static void main(String[] args) {
        String str = "hey!";
        boolean bool = false;

        System.out.print( (3 > 4) || (8 < 6) );
        System.out.print( (9 > -9) && (3 > 6) );
        System.out.print( !(6 < 5) && (34 >= 0) );
        System.out.print( !bool && str.equals("hey!") );
        System.out.print( bool || (3 < -6) || str.equalsIgnoreCase("HEY!") );
    }
}
```

20. Which of the following are **not** benefits of inheritance? Circle all that apply (4 points)

- a. Helps to eliminate repetition of code between classes
- b. Eliminates the need to create objects
- c. Polymorphism
- d. All fields automatically become public

21. Using the IS-A test, circle every item that makes sense as a subclass to a class called Plant. (6 points)

Tree  
Animal

Dog  
Flower

Grass  
LivingThings

22. Write a method that identifies some basic shapes. The method takes in two parameters: one for the number of sides of the shape, and another to indicate whether the sides are all equal length or not. The method should be able to identify and return the following shapes based on the provided parameter values: (10 points)

	numSides	equalLength
Circle	1	true or false
Triangle	3	true or false
Rectangle	4	false
Square	4	true
Unknown	Any other number	true or false