

1. The following questions are true or false. Please write a T or F on each line. (5 points)

- a. The **final** keyword indicates that a variable is a constant unchanging value.
- b. The **static** keyword means you need to have created an object to call the method.
- c. The **void** keyword is used to note when a method returns an empty String.
- d. The **throws** keyword is used to catch and handle an exception.
- e. The **extends** keyword is used to create an inheritance relationship between two classes.

T
F
F
F
T

2. What is the difference between a class (instance) variable and a local variable? (2 points)

- a. A class variable only exists within the scope of the method, while a local variable exists within the scope of the entire class
- ☒ b. A class variable exists within the scope of the entire class, while a local variable only exists within the scope of the method
- c. Class variables must be public, while local variables must be protected
- d. Class variables must be protected, while local variables must be public

3. Why is encapsulation good practice? (2 points)

- a. It ensures that a class's methods are completely inaccessible
- b. It allows the programmer to be able to create objects for a class
- c. It allows a class to be able to read and write to a file
- ☒ d. It ensures that a class's fields are only accessed in a controlled way

4. Match the terms to their definitions. (10 points)

i class

e object

j primitive type

c polymorphism

d constructor

f relative path

b absolute path

h file buffer

a file handle

g new

a. A reference to the file buffer

b. The entire path from the root directory

c. The idea that a class can be referred to by any of its parent classes

d. Used to initialize fields of a class; called when an object is created

e. A specific instance or version of a class

f. The path specified from a default working directory

g. Used to indicate object creation and set aside memory

h. An area in memory where file contents get copied to for faster access

i. A template or outline to represent a generic idea or concept

j. Basic data types stored as values rather than references

5. Use the code snippet to answer the following questions.

```
public class Final {  
    public static void main(String[] args) {  
  
        String[][] contactsPerPerson = { {"Jim", "Mark", "Anne"},  
            {"Kim", "Dave", "Adrian", "John"},  
            {"Meghan", "Sarah"} };  
  
        for (int i = 0; i < contactsPerPerson.length; i++) {  
            #1 System.out.println("Person #" + i + "'s contacts:");  
            #2 System.out.println(contactsPerPerson[i]);  
            for (String contact : contactsPerPerson[i]) {  
                #3 System.out.println(contact);  
            }  
        }  
    }  
}
```

a. When $i = 0$, what does line #1 print out? (2 points)

Person #0's contacts

b. When $i = 0$, what does line #2 print out? (2 points)

mem addr

c. When $i = 0$, what does line #3 print out? You do not need to worry about newline characters. (2 points)

Jim Mark Anne

d. What indices do I need to access the name "Adrian" from contactsPerPerson? (2 points)

contactsPerPerson[1][2]

e. Is the initialization of the contactsPerPerson array allowed in Java? (2 points)

Yes No

f. What is a benefit of using ArrayLists instead of a two-dimensional array to store the contact data? (2 points)

- i. The size would be dynamic, and you could add additional contacts easily
- ii. ArrayLists are not objects
- iii. ArrayLists are always smaller than arrays
- iv. The need for loops is eliminated when using an ArrayList

6. Use the UML for a class called ComputerScientist to help answer the following questions:

- a. Write a constructor for the ComputerScientist class that takes in three parameters and initializes all the class fields to the values passed into the constructor. (5 points)

```
public ComputerScientist(char g, String  
n, boolean iA){  
    setGender(g);  
    setName(n);  
    setAlive(iA);  
}
```

ComputerScientist

```
-gender : char  
-name : String  
-isAlive : boolean  
  
+getGender() : char  
+setGender(gender : char) : void  
+getName() : String  
+setName(name : String) : void  
+isAlive() : boolean  
+setAlive(alive : boolean) : void
```

- b. Write one line of code that creates a ComputerScientist object using the above constructor. You can use any values that you want for the parameters. They do not have to match with an existing computer scientist. For the gender, just use the first letter of the person's gender. (3 points)

```
ComputerScientist cs = new ComputerScientist('f', 'Ada', false);
```

- c. Fill in the if statements for the following code that tries to guess the identity of the ComputerScientist object you created in part b. Use the comments to help you identify what needs to be checked in each statement. Make sure to utilize the methods defined in the ComputerScientist class. (4 points)

```
public class Final {  
    public static void main(String[] args) {  
        String guess = "";  
        // check if the ComputerScientist is female  
        if ( cs.getGender() == 'f' ) {  
            // check if the ComputerScientist is alive  
            if ( cs.isAlive() ) {  
                guess = "Barbara Liskov";  
            } else {  
                guess = "Grace Hopper";  
            }  
        }  
        // check if the ComputerScientist is male  
        } else if ( cs.getGender() == 'm' ) {  
            // check if the ComputerScientist is alive  
            if ( cs.isAlive() ) {  
                guess = "James Gosling";  
            } else {  
                guess = "Alan Turing";  
            }  
        }  
    }  
}
```

d. Write a toString() method for the ComputerScientist class in the space below (5 points):

```
public String toString() {  
    return "CS: " + name + "(" + gender + ")" + isAlive;  
}
```

- e. What concept is demonstrated when writing a toString() method? (2 points) overriding
- f. Suppose the following methods are also defined in the ComputerScientist class. What concept does this demonstrate? (2 points)

```
public boolean checkIfSameGender(ComputerScientist cs) {  
    if (cs.getGender() == this.getGender()) {  
        return true;  
    }  
    return false;  
}
```

```
public boolean checkIfSameGender(char gender) {  
    if (this.getGender() == gender) {  
        return true;  
    }  
    return false;  
}
```

overloading

7. Circle all the following **true** statements about “pass by value” and “pass by reference”. (4 points)

- ☒ a. Pass by value applies to primitive types
- ☐ b. Pass by value applies to objects
- ☐ c. Pass by value means that the value of the variable can be modified by the method
- ☒ d. Pass by value means that a copy of the variable is used within the method
- ☐ e. Pass by reference applies to primitive types
- ☒ f. Pass by reference applies to objects
- ☒ g. Pass by reference means that the value of the variable can be modified by the method
- ☐ h. Pass by reference means that a copy of the variable is used within the method

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

RussEngineering

Airport

OelmanHall

StudentUnion

Building

Grass

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

```
public class Final {  
    public static void main(String[] args) {  
        int value = -8;  
        boolean isValid = true;  
        String greeting = "hi";  
  
        System.out.print((6 > 4) || isValid);  
        System.out.print(((value - 3) < -9) && !isValid);  
        System.out.print(!(9 > -3) || greeting.equalsIgnoreCase("HI"));  
        System.out.print(('f' != 'g') && (3.6 < value));  
        System.out.print(greeting.charAt(0) == 'h');  
    }  
}
```

T F T F T

10. Fill in the print statement of the ArrayList after the following code executes: (3 points)

```
public class Final {  
    public static void main(String[] args) {  
        int[] totalTimes = {43, 37, 39, 46, 41, 55};  
        ArrayList<Boolean> deliveredOnTime = new ArrayList<>();  
  
        for (int time : totalTimes) {  
            if (time < 45) {  
                deliveredOnTime.add(true);  
            } else {  
                deliveredOnTime.add(false);  
            }  
        }  
        System.out.println(deliveredOnTime);  
    }  
}
```

[T T T F T F]

11. Match the part of the method header to its label. Some labels may be used twice (5 points)

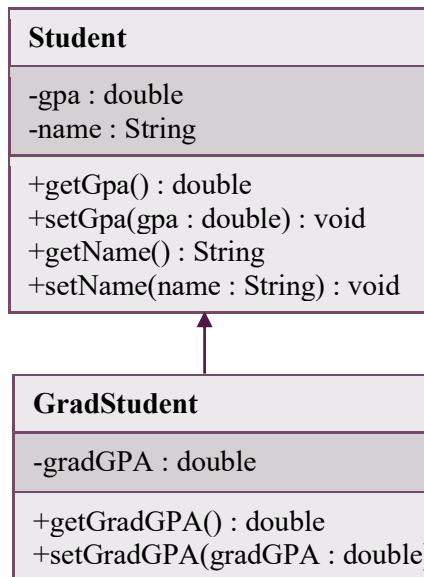
```
public static double convertTemp(double tempDegF, double tempDegC)
```

1. method name 2. parameter name 3. return type 4. access modifier 5. parameter data type

12. Write a method that counts and **returns** how many times the word “the” appears with any capitalization in an ArrayList of String. (10 points)

```
public static int countThe(ArrayList<String> in) {  
    int cnt = 0;  
    for (String s : in) {  
        if (s.equalsIgnoreCase("the")) {  
            cnt++;  
        }  
    }  
    return cnt;  
}
```

13. Using the provided UML, circle all the following that are valid statements. (6 points)



- ☒ a. GradStudent gs = new Student();
☐ b. Student gs = new GradStudent();

For the next few statements, assume a Student object called stu has been successfully created, and a GradStudent object called gStu has been successfully created.

- ☒ a. stu.setGpa(4.0);
☒ b. gStu.setGpa(3.5);
☐ c. stu.setGradGPA(3.6);
☒ d. gStu.setGradGPA(3.8);

14. Which of the following are objects? Circle all that apply. (4 points)

- ☒ a. String
☐ b. int
☒ c. double[]
☐ d. float

15. What happens if you try to print out an object that does not have a toString() defined? (2 points)

- ☐ a. It will print out all the object's fields in a list
☐ b. An error will be thrown
☒ c. The memory address will be printed out
☐ d. Nothing will be printed out

16. What does the % operator do? (2 points)

- a. Calculates the percent of a number
- ☒ b. Determines the remainder of dividing two numbers
- c. Rounds the number to the nearest whole value
- d. Generates a random number

17. What does the ArrayList set() method do? (2 points)

- ☒ a. Replaces the value at the provided index with the provided value
- b. Creates a new ArrayList
- c. Creates a HashSet for the ArrayList
- d. Appends the provided value to the end of the ArrayList

18. What is the purpose of the super keyword? (2 points)

- a. It is used to refer to variables and methods in the current class
- b. It creates a parent class file for the current class
- c. It integrates ChatGPT into the code
- ☒ d. It is used to refer to variables and methods of the parent class

19. What is the purpose of the this keyword? (2 points)

- ☒ a. It is used to refer to variables and methods in the current class
- b. It creates a parent class file for the current class
- c. It integrates ChatGPT into the code
- d. It is used to refer to variables and methods of the parent class