**HW07 Arrays and Objects**

Due Date: **See Canvas**

Purpose: Demonstrate understanding of working with methods and arrays.

Effort: **Individual**: Read [CS Academic Integrity .pdf](https://drive.google.com/file/d/1vHXGQPPRUXZQrUxfnM6hJQid9jio7xlt/view?usp=sharing)

Points: **100 (see rubric in canvas)**

Deliverables: **Upload your design document with your design/ reflection/learning question and your .java file as separate files. Do not upload as a zip file.**

**Assignment Description**

This assignment provides the opportunity to work with an array of objects and an object that contains an array. You will create several **turtle** objects and then perform two tasks:

* Task #1: place all turtle objects into an array and perform manipulations on the array in main.
* Task #2: create one **turtle hospital** object, place all turtle objects into the **turtle hospital**, and perform similar manipulations on the array that is now in the turtle hospital.

These two tasks provide the opportunity to see the difference between working with an array declared in main and working with an array declared in a class as a private instance variable (HAS-A relationship).

In this assignment, the completed program contains **three classes:**

* **Assignment10** – contains main (test program that creates and works with the other 2 classes)
* **Turtle** – represents a turtle
* **Turtle Hospital** – represents a very small turtle hospital

Details for both **Turtle** and **Turtle Hospital** classes are provided in the “Classes” section below.

**Specifications**

Read these specifications to get an overview of assignment. When writing code, **first** create the classes.

1. Create a Java class **LastNameFirstNameHW07** within that project calledwithin the **CS1150HW** project
2. Follow [Problem Solving Using Software Development Life Cycle](https://docs.google.com/presentation/d/e/2PACX-1vSC4tM0BsOaYjRmd_emNeHtKrWaExKXfvvLxQC10rCt77CPT_WvF_s3W--o_yuj8XczYGo7qHxgjMIq/pub?start=false&loop=false&delayms=3000&slide=id.ge6997ea817_0_125)
3. Design algorithms for ***main*** and ***findOldestTurtle***method in Turtle Hospital class

* Most of the methods within the classes are small so there is no need to write pseudocode for the classes.

1. Write a test program (main) that:
   1. Creates the following 7 **Turtle** objects with the following **type, age,** and **name.**
      1. These turtle objects will be used in Part 1 and 2 below.

Hard code these values in the call to the constructor because it’s too much information for a user/grader to enter. Please use the provided information.

| **Turtle Object** | **Type** | **Age** | **Name** |
| --- | --- | --- | --- |
| 1 | Hawksbill | 42 | Rose |
| 2 | Leatherback | 30 | Donna |
| 3 | Loggerhead | 62 | Pancake |
| 4 | Greenturtle | 39 | Olive |
| 5 | Loggerhead | 75 | Poptart |
| 6 | Hawksbill | 33 | Miss Piggy |
| 7 | Leatherback | 14 | B StreiSAND |

* 1. **Part 1:**
     1. Create an array that will hold the **Turtle** objects
     2. Add each turtle to the array
     3. Print each turtle in the array
        1. Use a **for loop** to iterate through the array of turtles
        2. Obtain type, age, and name from the turtle object in the array
        3. Print the **type**, **age,** and **name** of the turtle
     4. Find the oldest turtle in the array
        1. Use a **for loop** to iterate through the array of turtles
        2. Determine which turtle is the oldest
        3. Print the **type**, **age,** and **name** of this turtle
     5. See output section below for an example of the output for part 1
  2. **Part 2:**
     1. Create one **Turtle Hospital** object
        1. Send the constructor the hospital name and the max number of turtles that the hospital can hold – for this assignment set the max to 9 turtles
     2. Note when working with **Turtle Hospital**:
        1. The code in main **does not have access to the array** inside the hospital object since the array is a private instance variable in the hospital class.
        2. To work with the “hidden” array, use the methods provided by the **Turtle Hospital** class.
     3. Add turtles to the **Turtle Hospital**
        1. First, add the turtles created in step 1 to the hospital
           1. Use the ***addTurtle*** method provided in **Turtle Hospital**.
           2. Be sure to display in main if each addition of a turtle is successful.
           3. See output below.
        2. Second, create 3 new turtles and add these turtles to the hospital
           1. Use the ***addTurtle*** method provided in **Turtle Hospital** class.
           2. Here are 3 new turtles:

| **Turtle Object** | **Type** | **Age** | **Name** |
| --- | --- | --- | --- |
| 8 | Greenturtle | 150 | Crush |
| 9 | Leatherback | 18 | Myrtle |
| 10 | Hawksbill | 44 | Coco |

* + - * 1. Be sure to display in main if each addition of a turtle is successful.
        2. See output below.
    1. Print each turtle in the hospital
       1. Use ***displayTurtles*** method provided in **Turtle Hospital** to print the turtles
       2. Print the **type**, **age,** and **name** of each turtle in the method
    2. Find the oldest turtle in the hospital
       1. Use ***findOldestTurtle*** method provided in **Turtle Hospital** to find oldest turtle
       2. Print the **name** of the hospital
       3. Print the **type, age,** and **name** of the oldest turtle in main
    3. See output section below for an example of the output for part 2

**Classes**

Design two classes, **Turtle** and **Turtle Hospital**, that contain the following data fields (instance variables) and methods.

**Turtle Class**

* Description
  + Class that represents one turtle
* Private Data Fields (all fields must be defined as **private**)
  + **type –** string identifier for the turtle’s type
  + **age –** integer identifier for the turtle’s age
  + **name –** string identifier for the turtle’s name
* Public Methods
  + Constructor

***public Turtle (String type, int age, String name)***

* + - Initializes instance variables **type, age,** and **name** to incoming type, age, name
  + Getters
    - Create getter for instance variables **type, age,** and **name**
  + Setters
    - None

**TurtleHospital Class**

* Description
  + Class that represents a turtle hospital
  + The hospital can keep only a certain number of turtles
* Private Data Fields (all fields must be defined as **private**)
  + **hospitalName** – string identifier for hospital’s name
  + **turtles –** array of turtle objects
    - Represents the turtles in the hospital.
    - This is only the declaration of the array, **not** the allocation of the memory.
    - The size for the array is given when a Turtle Hospital object is created, that is, in the constructor, so memory allocation occurs in constructor.
  + **numTurtlesInHospital –** integer for number of turtles currently in array
    - Keeps track of how many turtles are currently in the hospital’s array.
    - Use as the index into the array of turtles when adding turtles and finding oldest. Since numTurtlesInHospital is initialized to zero, the 1st time a turtle is added, using **numTurtlesInHospital** as the index into array will place the turtle into the 0th location in array.
    - Incremented each time a turtle is added to the hospital.
* Public Methods
  + Constructor

***public TurtleHospital(String hospitalName, int maxNumTurtles) { ... }***

* + - Initialize instance variable **hospitalName** to incoming value.
    - Allocate memory for the **turtles** array with the following code:
      * The code is provided to show you how to allocate the memory.
      * Use incoming **maxNumTurtles** for the size of the array
      * Be sure you understand what this code is doing.

turtles = **new** Turtle[maxNumTurtles];

* + - Initialize **numTurtlesInHospital** to zero since there are no turtles in the hospital (in the array) when the Turtle Hospital object is first created.
  + Getters
    - Create getter for instance variable **hospitalName**
  + Setters
    - ***public boolean addTurtle(Turtle turtleToAdd) { ... }***
      * If the array is not full, add the incoming turtle to the hospital’s array of turtles at array location **numTurtlesInHospital**.
      * The method must ensure the hospital array is not full.
      * The method increments the instance variable **numTurtlesInHospital** when turtle is successfully added
      * Returns boolean indicating if turtle was successfully added
  + Other
    - ***public Turtle findOldestTurtle () { ... }***
      * Returns the oldest turtle in hospital’s array of turtles
      * Note the return type is a **Turtle** object.
      * This method does not print the turtle details, it only finds the oldest turtle and returns that turtle to the caller.
      * Note that the return type is **Turtle**.
    - ***public void displayTurtles() { ... }***
      * Displays all turtles in the hospital’s array of turtles
      * Displays each turtle’s **type**, **age**, and **name**

**Specifications**

* In the classes, Turtle and TurtleHospital, all **instance variables** must be **private**.
* Please place all classes into **one Java file**
  + **Assignment10 class, Turtle class, and Turtle Hospital class**
* To do so, the layout should be as follows:

**public** **class** Assignment10 {

**public** **static** **void** main(String[] args) {

// Code to create turtle objects and turtle hospital object goes here

} //main

} // Assignment10 Note this is the end of your assignment 10 

New classes go **AFTER** this point

New classes **do not** include public keyword

**class** Turtle {

// Details for the turtle class goes here

} // Turtle

**class** Turtle Hospital {

// Details for the turtle hospital class goes here

} // Turtle Hospital

**Tip: No information is needed from the user.**

* Turtle types, ages, and names are **not** obtained from user.
* Use the values provided in the table in step 4a and write these directly into your code.
* **This is hard coding but it is fine for this assignment because:** 
  + There is too much information for the user to enter (think of the grader!)
  + We haven’t learned to use files yet.
* Soon, we’ll learn to read from files and do away with this type of hard coding.

**Output**

Your output should look similar to the following:

**Output - Example**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Part 1: Turtles In Array in Main

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--------------------------------------------

Turtle Age Name

--------------------------------------------

Hawksbill 42 Rose

Leatherback 30 Donna

Loggerhead 62 Pancake

Greenturtle 39 Olive

Loggerhead 75 Poptart

Hawksbill 33 Miss Piggy

Leatherback 14 B StreiSAND

--------------------------------------------

Oldest turtle in the array of turtles

--------------------------------------------

Type: Loggerhead

Age: 75 years

Name: Poptart

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Part 2: Turtles In Turtle Hospital

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Was Rose added to hospital? true

Was Donna added to hospital? true

Was Pancake added to hospital? true

Was Olive added to hospital? true

Was Poptart added to hospital? true

Was Miss Piggy added to hospital? true

Was B StreiSAND added to hospital? true

Was Crush added to hospital? true

Was Myrtle added to hospital? true

Was Coco added to hospital? False

--------------------------------------------

Turtle Age Name

--------------------------------------------

Hawksbill 42 Rose

Leatherback 30 Donna

Loggerhead 62 Pancake

Greenturtle 39 Olive

Loggerhead 75 Poptart

Hawksbill 33 Miss Piggy

Leatherback 14 B StreiSAND

Green Turtle 150 Crush

Leatherback 14 Myrtle

--------------------------------------------

Oldest turtle in turtle hospital

--------------------------------------------

Turtle Hospital: Key West Turtle Rescue

Type: Green Turtle

Age: 150 years

Name: Crush

## Learning/Reflections

Answer the following and include snippets of code from this assignment when explaining.

1. Explain how memory is allocated for an object on stack and heap. Include snippets of code and drawings with your explanation.

in DogShelter Dog1 = new dog();

DogShelter is declaring the object of DogShelter and creates memory on the stack to store an address

Dog1 = new assignes addresses referring to heap to dog1 variable

dog(); Allocaes memory on the heap at this address to store the object’s information.

1. Explain how memory is allocated for an array of objects on the stack and heap. Include snippets of code and drawings with your explanation.

new dog() is an object constructed with new operator memory which is allocated on the heap to store information on the dog() object.

DogShelter Dog1 sees memory allocated o the stack to store an address when the reference variable(dog1) is declared.

1. Explain how you access the age of a turtle in an array with code examples.

You can access the age of a turtle in an array by using a getter method. in the array turtles, I would use turtles[2].getAge.

1. Explain encapsulation and include code examples.

Encapsulation is used to hide certain parts of code to reduce the chance of errors. I made all instance variables private,

private String hospitalName;

private Turtle[] turtles;

private int numTurtlesInHospital;

and getters and setters are used to interface with these variables

//Getters for type age name

public String getType() {

return type;

}

public int getAge() {

return age;

}

public String getName() {

return name;

}

public int getNumber() {

return number;

}

1. Reflection:
   * What do you understand better after completing this assignment?

Manipulating arrays and using methods in objects.

* + What was the most difficult part of the assignment?

Trying to figure out the syntax of what I was trying to do and managing different data types and scopes.