

Name _____ Period _____

1. The Assignment class below creates and manages assignments for a gradebook,

Assignment.java

```
public class Assignment {
    public String name;
    public int totalPoints;
    public String dueDate;
    public boolean countInGrade;

    public Assignment(String n, int tp, String dd, boolean c){
        name = n;
        totalPoints = tp;
        dueDate = dd;
        countInGrade = c;
    }
}
```

A portion of the Gradebook class is shown below. You will write code to complete the remainder of this class.

Gradebook.java

```
import java.util.Scanner;

public class Gradebook{

    public static void main(String args[]){
        //Creates a gradebook with 5 assignments
        /* To be completed in part (a) */

        //Prompts the user for the assignment information
        Scanner input = new Scanner(System.in);
        System.out.println("What is the name of the assignment?");
        String name = input.nextLine();
        System.out.println("How many points is the assignment out of?");
        int points = input.nextInt();
        System.out.println("What is the due date (mm/dd/yy)?");
        String dueDate = input.next();
        System.out.println("Does this count towards the grade?");
        boolean counts = input.nextBoolean();
    }
}
```

- (a) Declare an array called gradebook that can hold 5 assignments.

/1

- (b) A scanner is used to get the input required for each assignment. In the space below, write code that could be used to create an Assignment using the input provided.

	/2
--	----

- (c) Below is a list of assignments that have been stored in the array gradebook,

Index	Name	Total Points	Counts Towards Grade	Due Date
0	Exam 1	18	True	09/01/23
1	Exam 2	12	True	09/08/23
2	Exam 3	17	True	09/15/23
3	Lab 2	20	False	09/15/23
4	Ticket Out the Door 4	5	True	09/02/23

In the space below, indicate how you could find the assignment worth the most points. Once you have located the assignment, print its corresponding information. For example,

Lab 2 is worth 20 points

	/4
--	----

(d) A gradebook can be visualized as a series of parallel arrays as follows. Where the values in each array represent the total points earned on the corresponding assignment.

	Exam 1	Exam 2	Exam 3	Lab 2	Ticket Out the Door 4
Bart	15	9	14	20	2
Homer	14	11	12	18	4
Wilma	12	12	9	17	5

```
int Bart[] = {15, 9, 14, 20, 2};  
int Homer[] = {14, 11, 12, 18, 4};  
int Wilma[] = {12, 12, 9, 17, 5};
```

The `avgGrades` array stores the average grade for each assignment as a percentage. For example,

```
int avgGrades[] = {75, 88, 68, 92, 73}
```

In space below, write an algorithm that does the following,

- Declares and initializes a the `avgGrades` array
- Computes the average grade on each assignment and stores the result as a percentage in the appropriate index. Each index in the `avgGrade` array should map to the correct assignment in gradebook array.

(e) A report card needs to be generated for each student. Consider the report card for Bart,

Exam 1: 15/18

Exam 2: 9/12

Exam 3: 14/17

Lab 2: 20/20*

Ticket Out the Door 4: 2/5

Final Grade: 77

In the space below, write code that could be used to create the output shown for Bart. Assignments marked with an asterisk are not included in the final grade calculation.

/6