

Your Name: _____ Student Number: _____

Lab 3a: to be done in class with a partner – due at the end of the lab

Note: Sample code appears at the end of the lab write-up (Appendix A) which can help you.

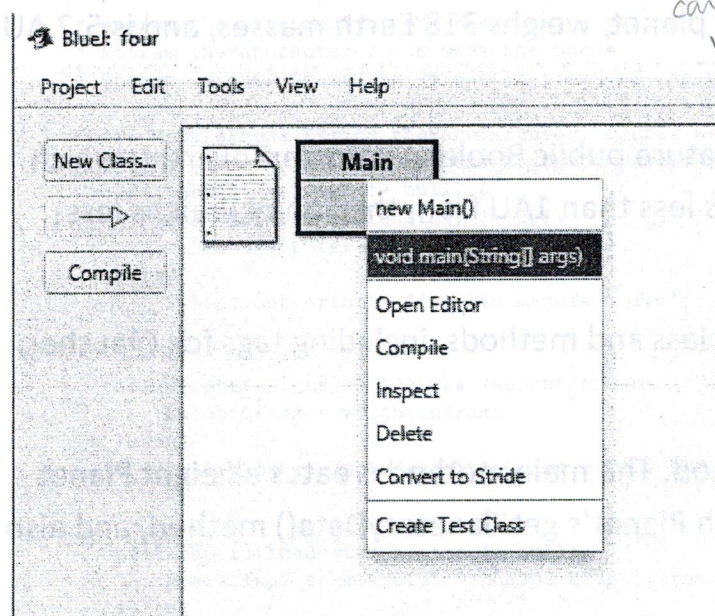
Book

Work with the Book class you created for Lab 2b. Address any comments from the previous submission. Then add accessor and mutator methods for all instance variables in the class.

Main

Create a new class called Main with a main method. Inside this main method, create a Book object (see the sample code in Appendix) with the title Harry Potter, written by JK Rowling in 1997. Use the Book's accessor methods to System.out.println the book's data in the format of "Harry Potter is 20 years old today, published in 1997"; use the String concatenation operator + to join the Strings together to be printed.

Note: to RUN this program, right-click on the Main class in BlueJ and select the main method:



Submission (10 marks)

This in-class lab is due at the end of this class. Do not upload your lab to BCIT's servers. When you are finished, show your instructor so he can sign your paper.

Checked by: _____

NOTE: keep this paper for your instructor to verify your grades later in the course.

NOTE: EVERY SINGLE STUDENT MUST SHOW THIS LAB AND GET HIS OR HER PAPER SIGNED....

Lab 3b: to be done at home alone – upload to D2L

Create a new class called Planet (e.g. https://en.wikipedia.org/wiki/Solar_System). It has instance variables, constructor parameters, and accessor methods for the following data:

- String name
- boolean innerPlanet *later*
- double astronomicalUnitsFromSun
- double earthMasses

Create a method called `getPlanetaryData()` which returns a String in the exact format of "Mercury is an inner planet, weighs 0.055 Earth masses, and is 0.4 AU from the Sun." or "Jupiter is an outer planet, weighs 318 Earth masses, and is 5.2 AU from the Sun.", etc....

Create another method with the signature `public Boolean isSimilarToEarth()` which returns true if and only if the Planet is less than 1AU from the Sun or weighs less than 15 Earth masses.

Make sure to properly JavaDoc your class and methods, including tags for `@author`, `@version`, `@param`, and `@return`.

Create a Main class with a main method. The main method creates all eight Planet objects for our solar system, calls each Planet's `getPlanetaryData()` method, and also

reports in one sentence for Planet whether the Planet is similar to Earth, in the format of "Jupiter is not similar to Earth."

Submission (15 marks)

Test your code and submit it to the Lab 3 folder on D2L (Activities -> Assignments -> Lab3) by Wednesday, Sept. 26, 2018 at midnight. The submission must include the **Planet.java** and **Main.java** files from your BlueJ project.

Mark Breakdown:

- (5 points) Compiles and can create an instance (in BlueJ)
 - 0 points if it does not compile (both Planet and Main must compile)
- (4 points) Valid JavaDoc (class and constructor), good names and data types
 - 0.5 points taken off for each violation (both Planet and Main)
- (6 points) Meets the specifications in the description
 - 4 points for Planet.java, 2 points for Main.java

Appendix A – Example Code

Example code which can help you with the general idea:

```
class MobilePhone{
    private String manufacturer;
    private double weightGrams;
    private String operatingSystemVersion;

    /**
     * @param theOperatingSystemVersion the os version
     * @param theManufacturer who made the phone
     * @param weightInGrams the weight in grams of the phone
     */
    public MobilePhone(String theManufacturer, double weightInGrams,
                       String theOperatingSystemVersion){
        if((theManufacturer != null) &&
            (theManufacturer.length() != 0)){
            manufacturer = theManufacturer;
        }else{
            System.out.println("invalid manufacturer");
        }

        if((weightInGrams >= 100) && (weightInGrams < 1000)){
            weightGrams = weightInGrams;
        }else{
            System.out.println(weightInGrams + " is not a valid weight in grams");
        }

        if(theOperatingSystemVersion != null){
            operatingSystemVersion = theOperatingSystemVersion;
        }else{

```



```

        System.out.println("null is not a valid os version");
    }
}

/**
 * @param newOperatingSystemVersion the new os
 */
public void setOperatingSystemVersion(String newOperatingSystemVersion){
    if(newOperatingSystemVersion != null){
        operatingSystemVersion = newOperatingSystemVersion;
    }else{
        System.out.println("null is not ok for os version");
    }
}

/**
 * @return the os version
 */
public String getOperatingSystemVersion(){
    return operatingSystemVersion;
}

/**
 * @return the manufacturer
 */
public String getManufacturer(){
    return manufacturer;
}

/**
 * @return the weight in grams
 */
public double getWeightGrams(){
    return weightGrams;
}

public static void main(String[] args){
    System.out.println("running our first program");
}

public void test(){
    System.out.println(5 + 5 + "5" + (5 + 5) + 5 + 5 + "(5+5)");
}

}

class Main{
    public static void main(String[] args){
        MobilePhone m1 = new MobilePhone("apple", 150, "10.0.1");
        MobilePhone m2 = new MobilePhone("samsung", 170, "5.5.5.5");

        System.out.println(m1.getOperatingSystemVersion());
    }
}

```