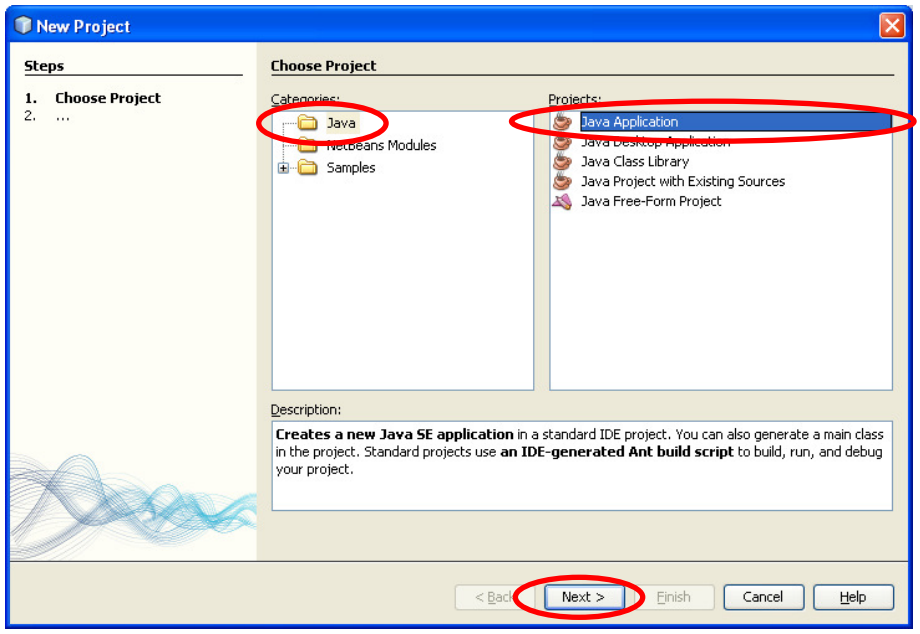
DePaul University
College of Computing and Digital Media

CSC 211 - Programming in Java I

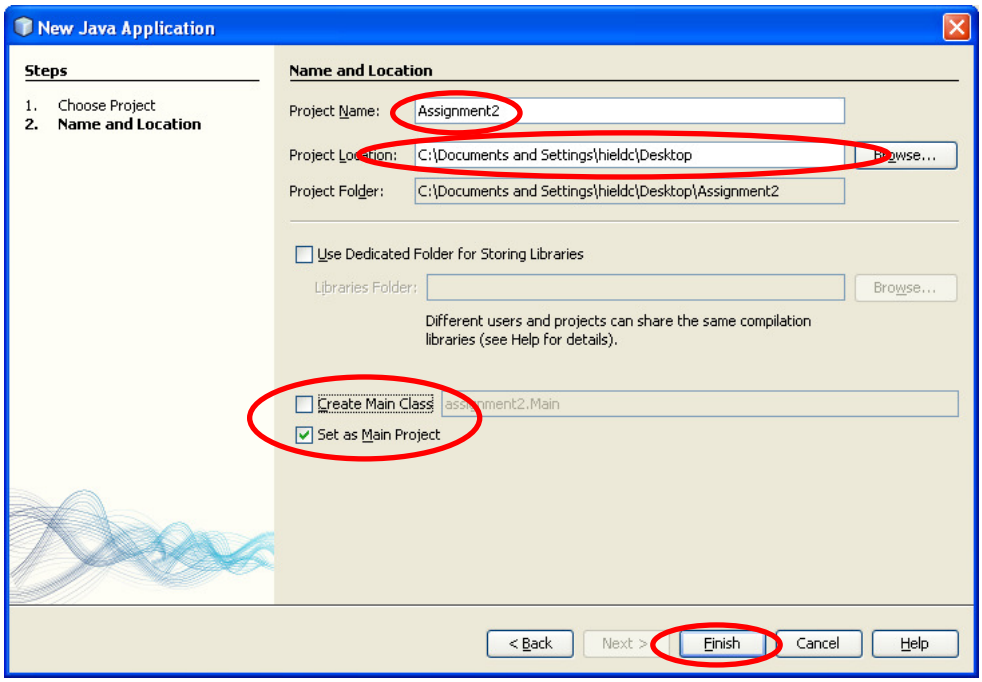
Assignment 2

Your assignment is to create a Java class that represents an Employee. To do this, you should do the following:

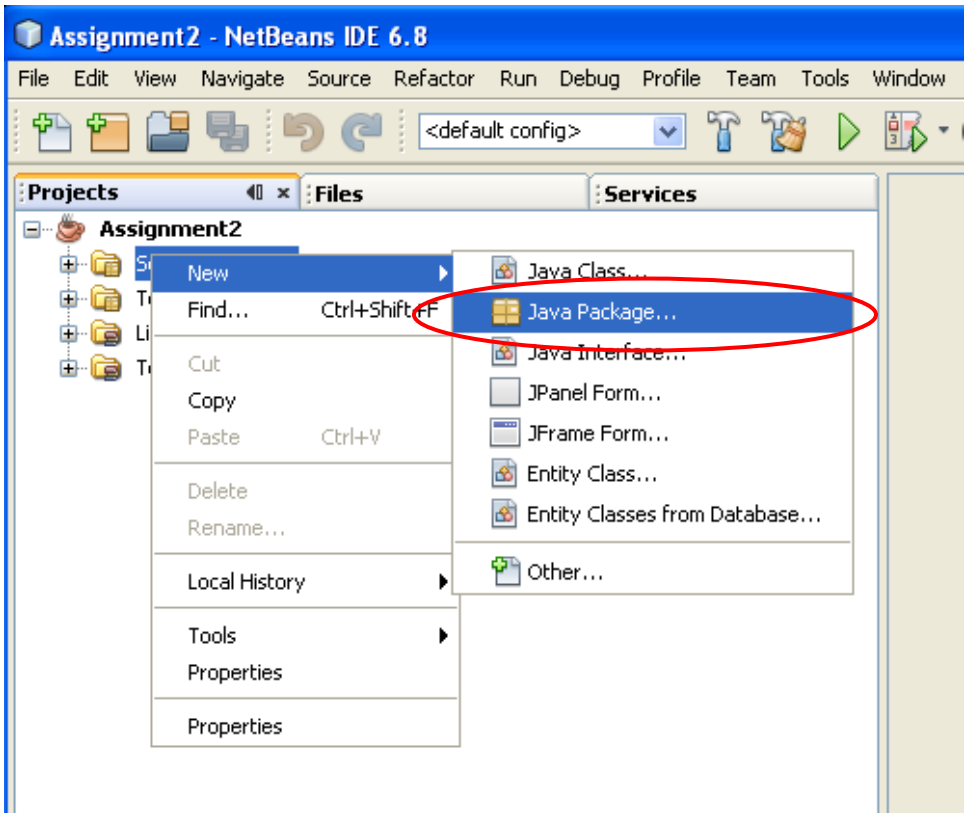
1) Create a new project in NetBeans (Java => Java Application)



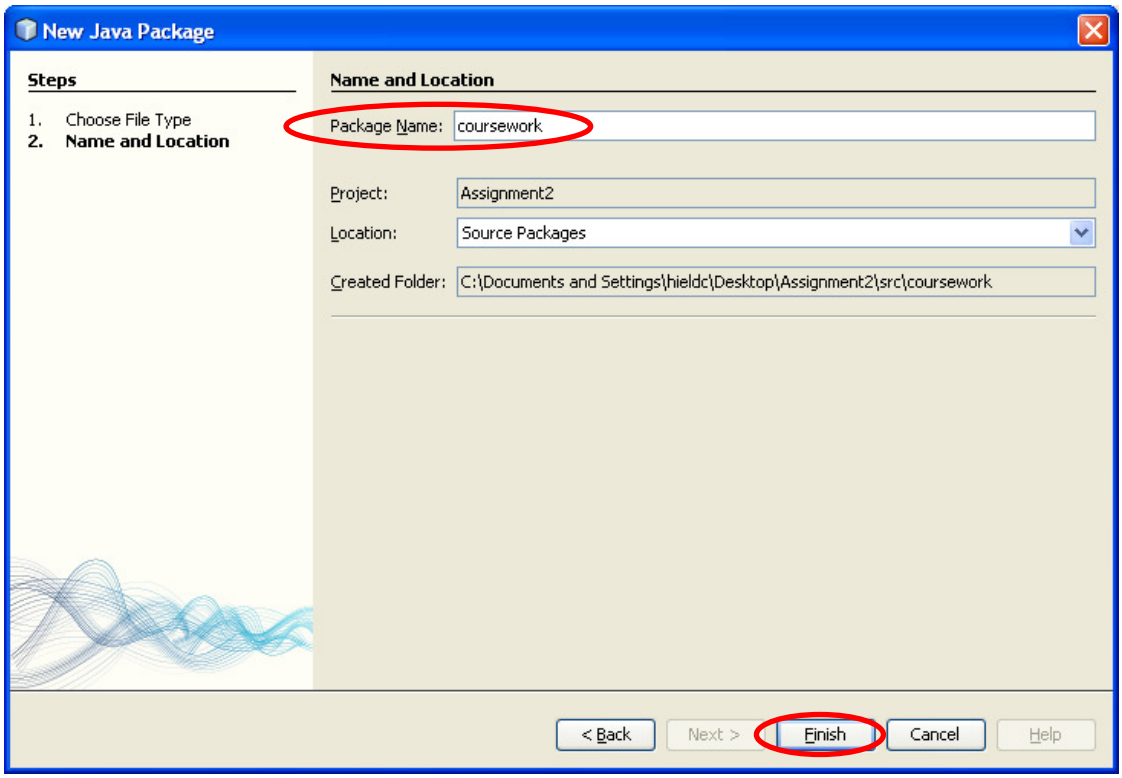
Name your project (i.e., "Assignment2"), and select a "Project Location" for your project folder *(I chose my Desktop in this example)*. Select the other options as shown below and then click "Finish". *(If the "Start Page" is displayed in the right-side editor, click on the "X" on the "Start Page" tab to close it.)*



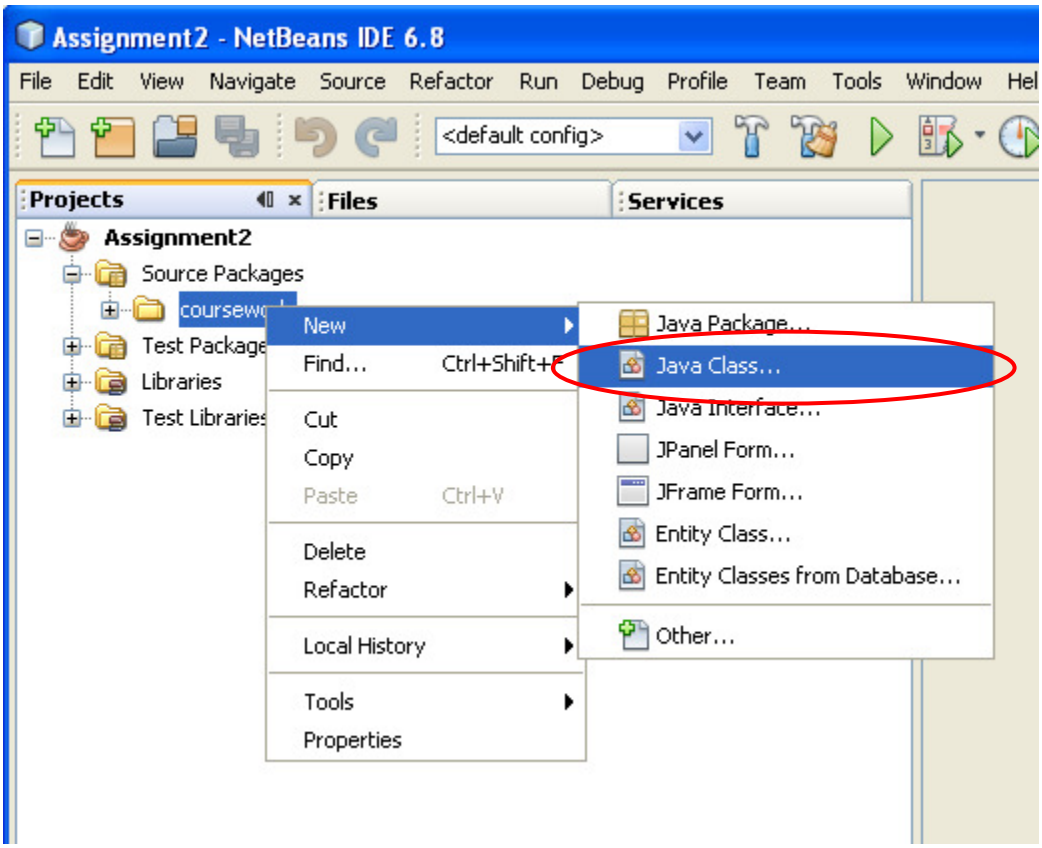
2) We will first create a java source "package" – the folder in which we will create our classes. To do this, right-click on the "Source Packages" entry in the NetBeans "Projects" tree-view and select *"New -> Java Package..."* as shown below:



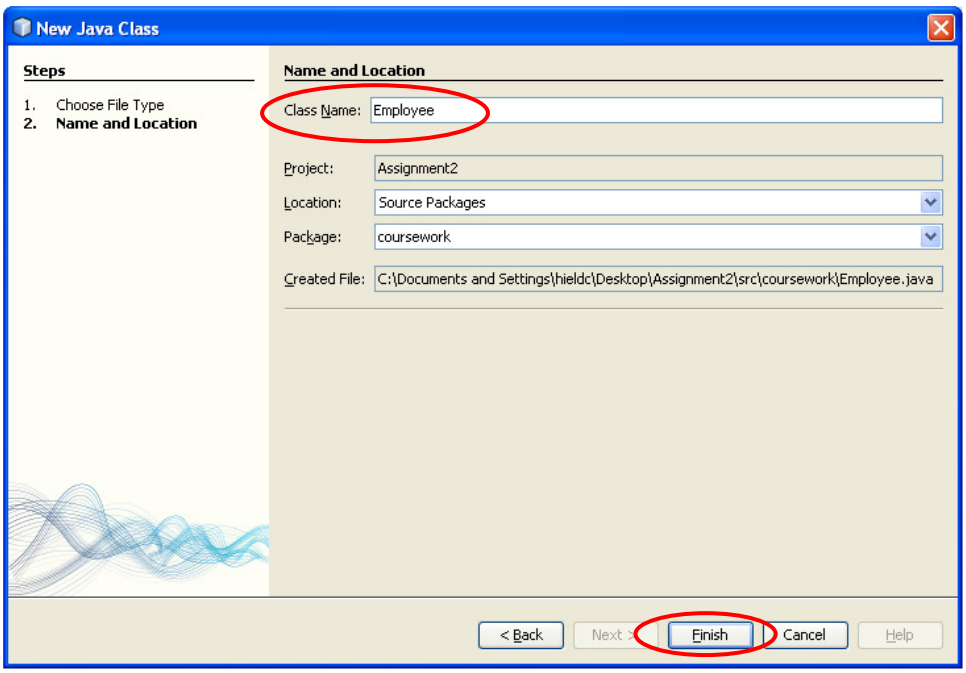
Call the new java package "coursework" and then click "Finish":



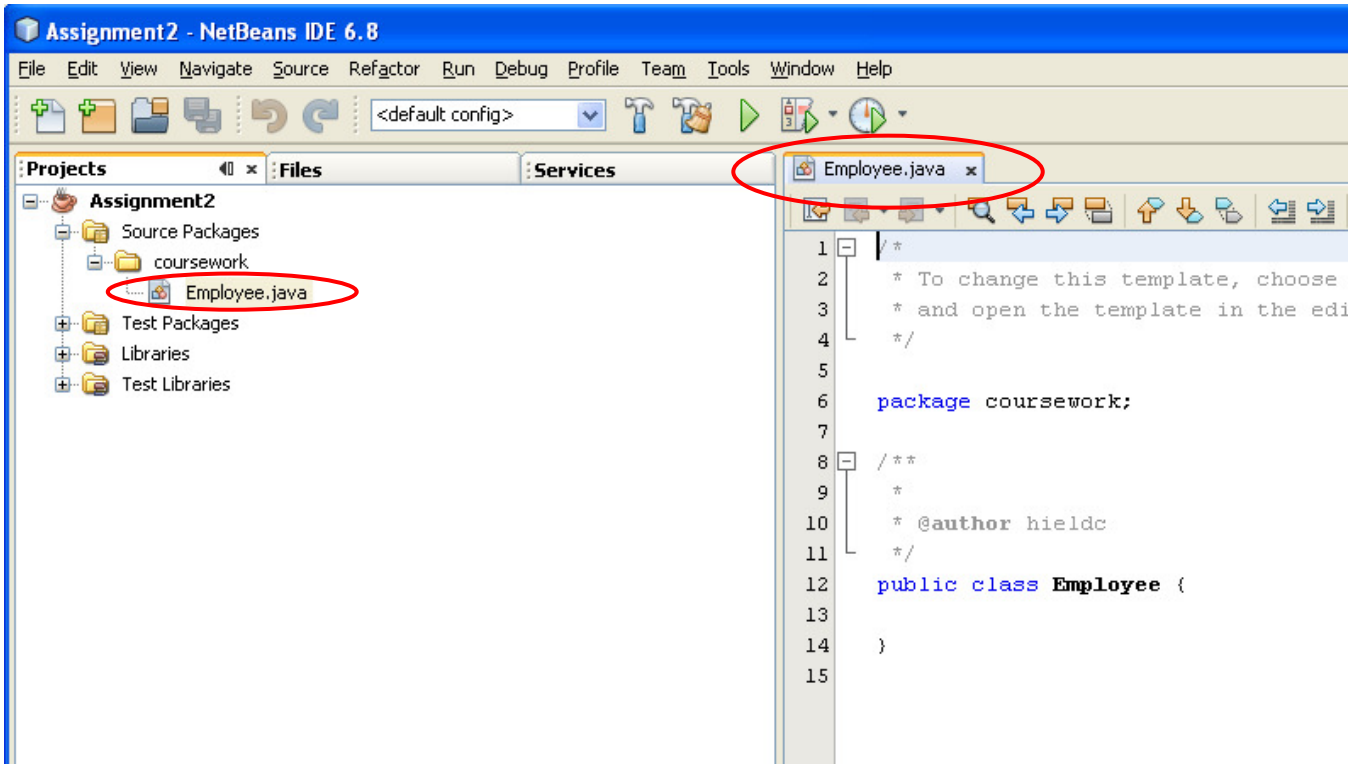
3) Now we want to create a new "Employee" class. To create the Employee class, right-click on your "coursework" package in the NetBeans "Projects" tree-view and select *"New -> Java Class..."* as shown below.



4) The "New Java Class" pop-up will appear. Enter "Employee" for the "Class Name". Then click "Finish".



5) The Employee class will be created, and its default contents will be displayed in the editor (on the right). The "Employee.java" file will also appear in the project contents list on the left.



6) Now, we will add some content to this new (empty) Employee class. First, add declarations for the following 4 *public* data members (the order in which they appear in your java class doesn't matter). Be sure to add these inside the Employee class (in between the opening curly-brace ("{") and the closing curly-brace ("}").

- A String to hold the Employee's name called "firstName".
- A String to hold the Employee's name called "lastName".
- An int to hold the employee's id number called "employeeId".
- A double to hold the employee's hourly rate called "hourlyRate".

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package coursework;

/**
 *
 * @author hieldc
 */
public class Employee {


}
```

Put your 4 data member declarations here!

7) We will continue editing the Employee class by creating a *constructor* that should be declared to new accept 4 parameters, one parameter for each of the data elements you created in the previous step. The constructor goes after your data member declarations. Remember – do not name constructor/method parameters using the same name as your data members.

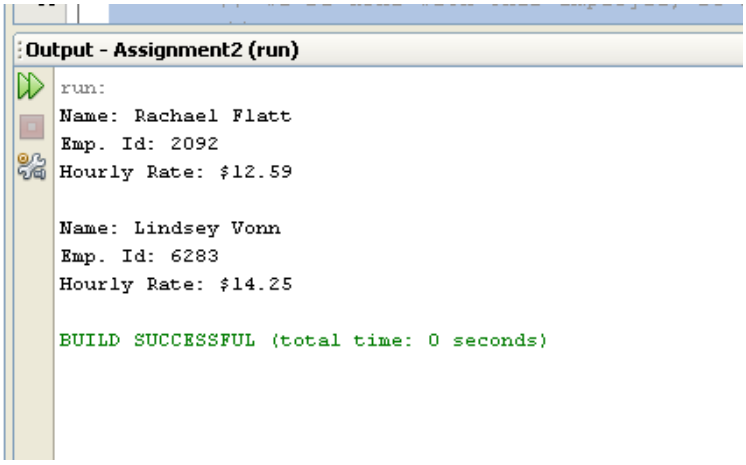
8) Within the constructor, use the 4 parameters you passed in to set the corresponding Employee data members.

9) Finally, *after* the Employee constructor you just created, add the following "main" method that will create 2 Employee objects and will print out their data member values to the screen:

```
public static void main(String[] args)
{
    // First we create a variable of type Employee and set it to a new
    // instance of an Employee, passing some test data to the constructor.
    Employee e = new Employee("Rachael", "Flatt", 2092, 12.59);

    // Print out the data with nice titles
    System.out.println("Name: " + e.firstName + " " + e.lastName);
    System.out.println("Emp. Id: " + e.employeeId);
    System.out.println("Hourly Rate: $" + e.hourlyRate);
    System.out.println(); // This will simply create a blank line

    // We're done with this employee, so now we'll re-use the Employee variable
    // called "e" by once again setting it to a new instance of an Employee,
    // passing different test data to the constructor.
    e = new Employee("Lindsey", "Vonn", 6283, 14.25);

    // Print out the data with nice titles
    System.out.println("Name: " + e.firstName + " " + e.lastName);
    System.out.println("Emp. Id: " + e.employeeId);
    System.out.println("Hourly Rate: $" + e.hourlyRate);
    System.out.println(); // This will simply create a blank line
}
```

10) Now compile your project – the "Employee.java" file will be compiled. Fix any compiler errors you may encounter. Then - run the program.

Expected output:



Submission:

- This assignment is due before the start of class next week (on or before 5:45 pm on Monday, April 19th). Late assignments will be penalized 10% per week.
- Your submission should consist of your entire project folder put into a single ZIP file (or a "TAR" file, or a "RAR" file). Check with me on other formats. *Your project folder is the folder you specified at the beginning of this assignment - in the window at the bottom of page #1 ("Project Folder:")*
- All submissions are to be made via the course's Course OnLine site
- You may email me with any questions on this assignment at any time between now and the due date.