# *Programming III (420-B31-HR)*

# *Lab 1 – Semester Start-up and Java Review*

Date assigned & due: Tuesday, August 23, 2016

**Objectives:**

1. Set up course folders.
2. Set up Outlook on the desktop.
3. Enter a timetable in Outlook.
4. Review:
   1. Creating abstract classes and methods
   2. Creating a subclass
   3. Overriding a superclass method
   4. Creating and using arrays
5. Learn to:
   1. Override the **toString()** and **equals()** methods of the **Object** class.

**To Be Handed In:**

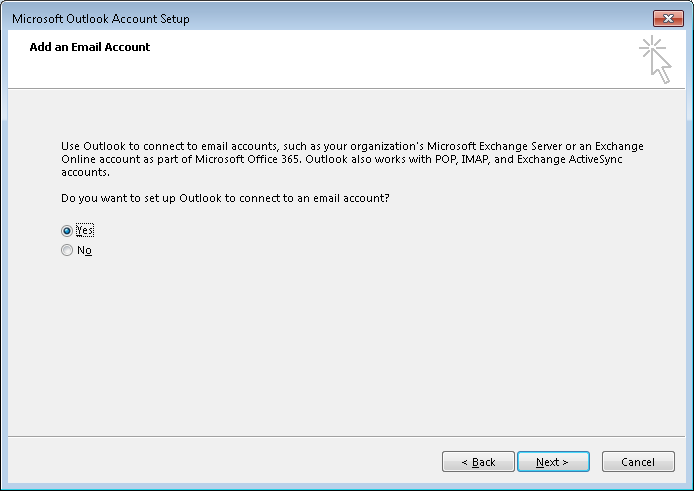
1. The ***username*\_B31\_L01\_Intro** folder should be zipped and uploaded to **Moodle**.
2. Show the teacher your completed calendar and lab to be marked.

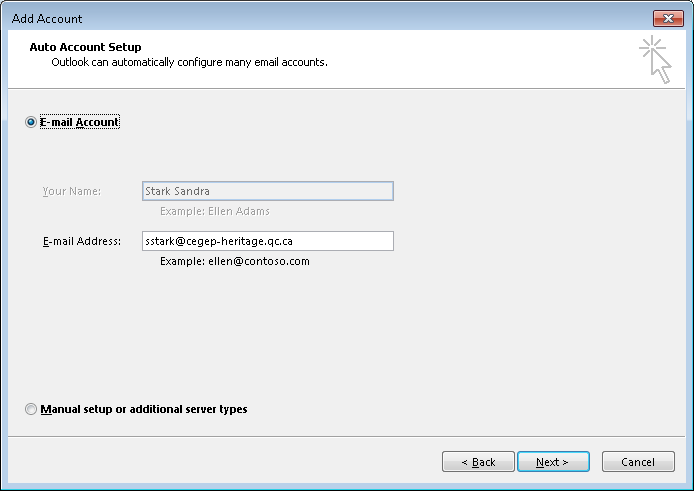
**Note: If you are in 420-D10 (Databases) then go directly to Part D. If you are not in 420-D10 then start at Part A.**

**Part A - Microsoft® Outlook Set Up**

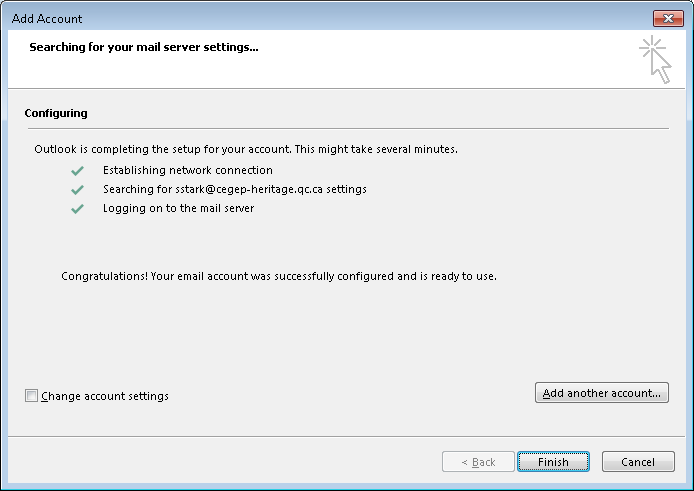
1. Start Microsoft Outlook 2013 by selecting **Start, Programs, Microsoft Office, Microsoft Office Outlook 2013** or using the icon in the task bar or desktop. The Outlook Startup window will open. Click Next.



1. The next window asks you to confirm that you want to set up an account. Make sure the Yes option is highlighted and click Next.  
     
   
2. The Add New Email Account screen is displayed. Your name and email address should be filled in by default. If your name is not automatically filled in, add your email address in the Email address field. Your email address is different this year: It is lastname.firstname@cegep-heritage.qc.ca.



1. Click **Next**. Your email account should be successfully set up now:



**If you do NOT receive the above screen, let me know immediately. Do NOT click finish. We may have to set up your account manually.**

1. Click **Finish**. Outlook 2013 will now open.

***Note***: You have set up your profile on the desktop computer. The next time you use this computer, simply open Outlook and select your profile.   
  
If you use another computer, you will have to repeat this procedure to set up your profile there.

**Part B - Calendar Set Up**

1. If it is not already started, start Microsoft® Outlook 2013 by selecting Start, Programs, Microsoft Office, Microsoft Office Outlook 2013 or using the icon in the task bar or desktop.
2. Select **Calendar** at the bottom of the screen and change the view to be **Week** by selecting the tab below the tool bar.
3. Set up your timetable by repeating the following steps for each course block in your timetable.
   1. Starting with Monday, August 22, select a block of time that corresponds to the course and then select **New Appointment** on the shortcut menu.
   2. Select the **Recurrence** option and ensure that the Appointment Recurrence is set up as follows:
      1. The Start and End time should be the starting hour and ending hour for the block in your timetable. The end time should not be changed to be 10 minutes before the hour. The Duration is automatically calculated.
      2. The Recurrence pattern should be weekly, and the appropriate day should be checked.
      3. Set the **Range of recurrence** to **End after: 16** occurrences, for courses on Monday, Tuesday or Wednesday and **15** for courses on Thursday or Friday. Click **OK**.
   3. In the **Subject** box, type the name of the course.
   4. In the **Location** box, enter the room number.
   5. In the top ribbon bar, set the Reminder (the bell icon) for the entry to be None.
   6. Click **Save and Close**.
4. Change your Outlook calendar to match the academic calendar as follows:
   1. Delete the occurrences on Labour Day (September 5) by selecting all the blocks on that day, selecting Delete from the shortcut menu, and ensuring that **Delete this occurrence** is selected.
   2. Delete the occurrences on the Wednesday after Thanksgiving (October 12) by selecting all the blocks on that day, selecting Delete from the shortcut menu, and ensuring that **Delete occurrence** is selected.

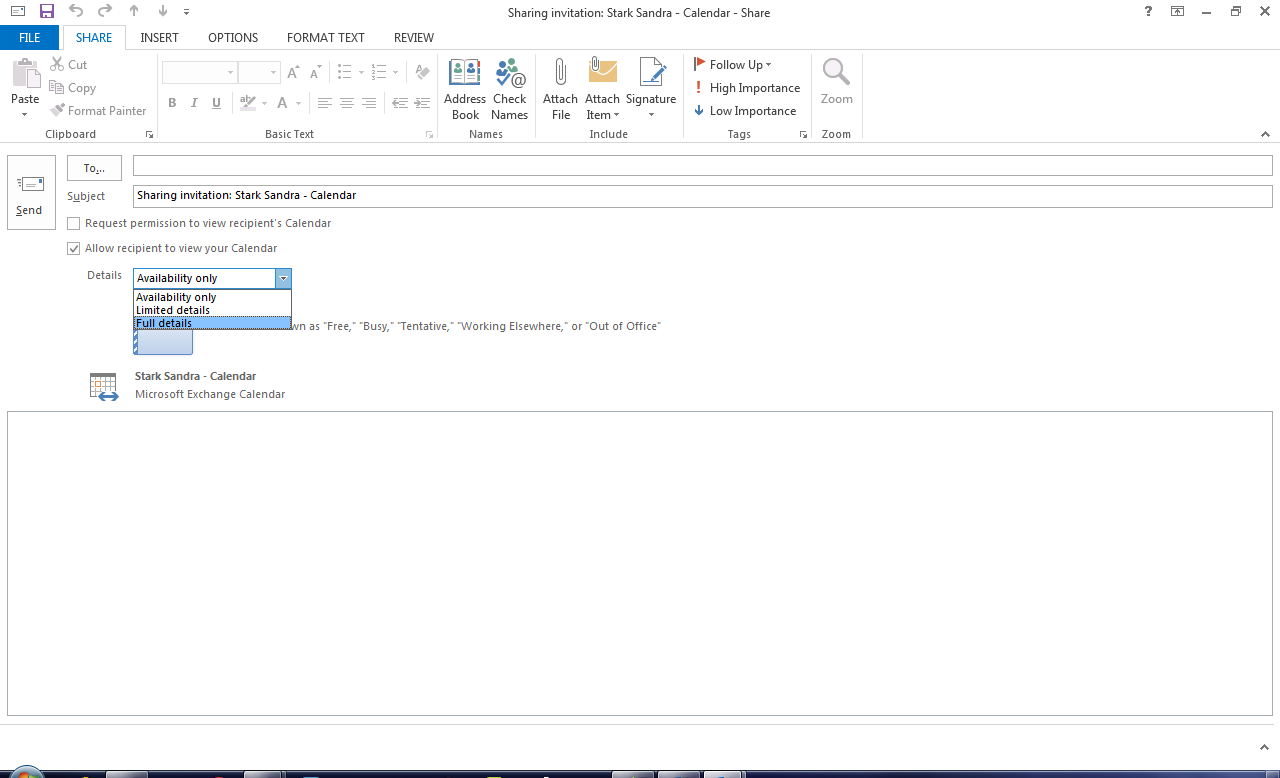
### Move the occurrences from Thanksgiving Monday (October 10) to the Wednesday after Thanksgiving (October 12) by double clicking the appointment and changing the date. Make sure that Open this occurrence is selected. You can also drag and drop the occurrence to the new day, in which case you will be prompted to confirm that you are changing only the occurrence. Select Yes.

* 1. Delete the occurrences on Tuesday November 1, by selecting all the blocks on that day, selecting Delete from the shortcut menu, and ensuring that **Delete occurrence** is selected.

### Move the occurrences from Friday November 4 (Study day) to Tuesday November 1 by double clicking the appointment and changing the date. Make sure that Open this occurrence is selected. You can also drag and drop the occurrence to the new day, in which case you will be prompted to confirm that you are changing only the occurrence. Select Yes.

1. While viewing your calendar click on the ShareCalendar button on the menu bar.

Share your calendar with the user **#Teaching Staff Computer Sciences** (all the Computer Science teachers). Under Details, select “Full Details”



Click **Send**.  
This will allow teachers to view your calendar.

**Part C: Set up Course Folders**

## Double-click on the My Computer icon at the upper left hand corner of the desktop. This opens Windows Explorer for the computer and network drives accessible to you.

## Double-click on the H: drive icon (*your\_username*). This is your home drive. It is also where My Documents will take you.

## Select File🡪 New 🡪 Folder from the menu bar.

## Replace the words New Folder with 420-B31.

## Double-click on 420-B31. Create two new folders here – call one Labs and the other one Assignments.

1. Your H: drive is tied to your student number this year. If you want to keep any of the data that is on your H drive from previous years, then you need to perform the following steps:
   1. log in to the computer using last year’s userid
   2. copy your data onto a USB stick or onto the C drive of the computer
   3. logout and then login using your student number
   4. copy your data from the USB stick or the C drive to your H: drive associated with your student number.
   5. If you used the C drive to copy the files, MAKE SURE YOU DELETE THE DATA FROM THE C: DRIVE!!

# Part D: Review of Inheritance

**Objectives:** Review the concepts of abstract class and method, inheritance and method overriding.

**To Do:**

## Copy the **B31\_L01\_Intro** folder in the **420-B31** folder from Moodle to your **420-B31\Labs** folder and rename it to ***username*\_B31\_L01\_Intro** folder.

## Start Eclipse.

## Select the ***username*\_B31\_L01\_Intro** folder as your workspace.

## Select **Import** from the **File** menu. Select **General -> Existing Projects into Workspace**. Select your ***username*\_B31\_L01\_Intro** folder. Your project should show up in the **Package Explorer**. Expand the **src** folder.

## The **Inheritance** package contains an **Animal**, **Pet**, **Lion**, **Dog** and **Cat** class. Modify the classes so that they match the inheritance hierarchy shown on the next page.

* **Animal** and **Pet** are abstract classes
* the **move()** and **speak()** methods in **Animal** are abstract
* **Pet** and **Lion** are subclasses of **Animal**
* **Dog** and **Cat** are subclasses of **Pet**

## Format each of the classes by right-clicking, and selecting **Source** 🡪 **Format**.



## Override any superclass methods that you need to in order to successfully compile the classes.

## Complete the constructors as follows:

| **Signature** | **Operation** |
| --- | --- |
| Pet() | the default superclass constructor is executed.  **petName** is set to “Unknown” |
| Pet(String petType) | **petType** is passed to the superclass constructor. **petName** is set to “Unknown” |
| Pet(String petType, char sex) | **petType** and **sex** are passed to the superclass constructor. **petName** is set to “Unknown” |
| Pet(String petType, String name, char sex) | **petType** and **sex** are passed to the superclass constructor.  **petName** is set to **name**. |
| Lion() | **“Lion”** is passed to the superclass constructor |
| Lion(char sex) | **“Lion”** and **sex** are passed to the superclass constructor |
| Dog() | **“Dog”** is passed to the superclass constructor |
| Dog(char sex) | **“Dog”** and **sex** are passed to the superclass constructor |
| Dog(char sex, String name) | **“Dog”**, **name** and **sex** are passed to the superclass constructor |
| Cat() | **“Cat”** is passed to the superclass constructor |
| Cat(char sex) | **“Cat”** and **sex** are passed to the superclass constructor |
| Cat(char sex, String name) | **“Cat”**, **name** and **sex** are passed to the superclass constructor |

## Create a **TestInheritance** class that contains a main method. It should create 2 Lion objects, 3 Dog objects and 3 Cat objects. Each object instantiation should use a different constructor. Run the program to test your inheritance hierarchy.

# Part E: Overriding the toString() and equals() Methods

**Objectives**: Learn to override the **toString()** and **equals()** methods of the Object class.

**To Do:**

## Add a **println** statement to **TestInheritance** to display the results of calling the **toString()** method for the **Lion**, **Cat** and **Dog** objects that are completely defined. (i.e. none of the attributes contain “Unknown”.)

## Run **TestInheritance**. What does the **toString()** method return?

Memory Locations

## Write the following overridden **toString()** methods:

|  |  |  |
| --- | --- | --- |
| **Class** | **Return value** | **Example** |
| Animal | the animal type and gender | “male horse” |
| Pet | the superclass toString() result concatenated with the name | “female hamster called Sarah” |

## Run **TestInheritance** again. What has changed?

No longer shows the memory location

## Create a **Dog** object with the same sex and name as the **Dog** object that is completely defined.

## Use the **equals()** method to compare the two identical **Dog** objects. Are they seen as equal?

## Override the **Object equals()** method in the **Animal** and **Pet** classes so that it yields the following results:

|  |  |
| --- | --- |
| **An object of** | **is equal to another object of same class if** |
| the **Animal** class | the **animalType** and **sex** are the same |
| the **Pet** class | the **animalType**, **sex** and **petName** are the same. Use the **equals()** method of the **Animal** class to see if the **animalType** and **sex** are the same. |

## To ensure that your **equals()** methods are working, add code to **TestInheritance** to execute the following tests:

* compare a **Dog** object with another **Dog** object with the different name
* compare a **Lion** object and with a **Cat** object
* Compare 2 Lions that have a different sex
* Compare 2 Lion objects that are identical.

## Make sure that all your classes are formatted before uploading your project to **Moodle**. Show the teacher your completed lab in order to have it marked.

# F. Moodle Quiz

1. Log on to **Moodle**, go to the **Programming III** course page and complete the **Lab 1 Terminology Quiz by Friday August 26**.

**Marking Scheme**

|  |  |  |
| --- | --- | --- |
| **Task** | **Mark** | **Out of** |
| Animal Class |  | 4 |
| Pet Class |  | 14 |
| Lion Class |  | 2 |
| Dog Class |  | 2 |
| Cat Class |  | 2 |
| TestInheritance Class |  | 20 |
| **Total** |  | **44** |