**chap 1 Getting to know Greenfoot**

**---** Install Java JDK and Greenfoot IDE, Run Java applications on Eclipse and Greenfoot

We use **Moodle** system to host online course materials via URL <http://online.occc.edu> . You start every lesson by reading the schedule document in each Moodle folder.

**Objectives**

* Understand grading procedures and class schedule by reading the course syllabus
* Install Greenfoot, and run the scenarios in textbook chapter 1
* Install **Java** JDK, and then install Java IDE (Integrated Development Environment) – “Eclipse Java EE”
* Describe the background and key components of Java
* Compile and execute a Java application
* Submit homework 1 to **Moodle drop box.**

**Schedule:** This lesson covers Moodle folder “chap 1”. Please follow the steps below.

1, read the **CS 2163 Course Syllabus** carefully, which can be found in Moodle “chap 1” folder.

2, install Java SE JDK 12. Please follow the instructions in the installation document “**InstallingJDK.docx**”, and we support Windows, Linux, and Mac OX. SE JDK stands for **Standard Edition** of **Java Development Kit**. Number 12 is the version, and we also refer it as the major version.

3, download and install Greenfoot IDE from this link: <http://www.greenfoot.org/download> , and it supports Windows, Mac OS X, and Linux. As of Jan, 2020, the Greenfoot version is 3.6.1.

4, download the scenario java source code files from textbook : click this link <http://www.greenfoot.org/book/> , and then the link “[Book scenarios](https://www.greenfoot.org/book/material/book-scenarios.zip) (23.5 Mb)” . After you get the downloaded zip file, unzip it and you will get the folder name “book-scenarios”. Click into this folder, and you will see all the textbook projects placed in chapter folders.

Please read the instructions below and learn how to zip and unzip files and folders. It is very important that you learn it so that you can submit your homework properly.

**zip / unzip instructions for Windows user:**

* To zip file(s) or folder(s), just select the file(s) or folder(s), and then right click the file(s) or folder(s), and select “Send To” 🡪 “Compressed (zipped) Folder”;
* To unzip, just right click the zip file and then select “Expand” 🡪 “Extract All” to extract it.

**zip / unzip instructions for Mac OS X user:**

* To zip file(s) or folder(s), just select the file(s) or folder(s), and then mouse right click, and then select 🡪 Compress “file or folder name”.
* To unzip file(s) or folder(s), just double click the select the zip file, and then the decompressed folder will show up in the same directory where the zip file locates.

**zip / unzip instructions for Linux user:**

* To zip file(s) or folder(s), use this command line:

**zip –r /zipFilePath/sample.zip File1.java File2.java folder1 folder2**

, where sample.zip is the name of zip file locates in /**zipFilePath**, followed by the list of file(s) folder(s) to be zipped.

* To unzip, use this command line:

**unzip /zipFilePath/sample.zip –d ./temp**

, where sample.zip is the name of zip file, and the destination folder to put the extracted files is a subdirectory name temp under the current directory. Don’t forget to put the dot symbol . that goes before the slash / symbol. If without the **–d ./temp**option, then the extracted files will be put into the current directory. If you only want to view the content of a zip file without extracting it, use this command:

**unzip –l /zipFilePath/sample.zip**

It is very **important** that you follow the above instructions to **create** zip file and **extract** zip file, because it **guarantees the compatibility** of the zip files generated across the three different platforms. You need to follow the above instructions when you create the zip file for your homework. **Do NOT** use **RAR zip files in this course,** and do **NOT** use **7z zip** **files** in this course.

5, How to open a greenfoot project? --- There are two different approaches:

Approach 1: start Greenfoot, and then choose menu item “Scenario”--> Open , and in the dialog box, locate the unzipped folder “book-scenarios” that you have unzip in the previous step, then locate subfolder “chapter01”, then locate the Greenfoot project scenario “leaves-and-wombats”, then click the “Select Folder” button to open the project leaves-and-wombats. To open a different project, just locate its corresponding folder and then click the “Select Folder” button, such as folder “asteroids1” under folder “chapter01”.

Approach 2: in Windows “File Explorer” or Mac Finder or Linux’s GUI File Manager, browse to folder

***book-scenarios--> chapter01 --> leaves-and-wombats***, and then click file ***project.greenfoot***, and you will see the Greenfoot application is fired up to open the project. You need to enable the file extension view in the GUI file tool, in order to see the file extension ***.greenfoot*** for the greenfoot project file. BTW, file ***project.greenfoot*** has a unique green foot icon, so you won’t miss it.

On how to show file extensions, refer to these links:

<http://kb.winzip.com/kb/entry/26/> is for Windows

<https://www.mactrast.com/2018/01/show-file-extensions-macos-finder/> is for Mac

As of Jan 2020, the current Greenfoot software version is 3.6.1. If you have either an older version of Greenfoot software or an older version of the scenario source file(if you download from step 4 above, then the scenario source code should be updated), you may experience this: when you open a scenario for the first time, if Greenfoot asks you whether to update the scenario source code or not, and you should click yes to do the update.

In case of source code compilation error in Greenfoot, please click the class name (the square box on the right panel of Greenfoot GUI, such as WombatWorld, Leaf, Wombat) to open its java source file(s), and then use // line comment symbol to comment out these two lines if they show up at the beginning of the source code file:

import java.awt.Color;

import java.awt.Font;

, so that it becomes:

//import java.awt.Color;

//import java.awt.Font;

Reason for commenting out these two lines:

This link <https://www.greenfoot.org/files/javadoc/> has all the Greenfoot classes listed, with a total of 9 classes including class Color and class Font. Greenfoot’s own class greenfoot.Color and greenfot.Font are not compatible with java’s standard library class java.awt.Color and java.awt.Font, therefore, we cannot import both greenfoot.Color and java.awt.Color at the same time, and that is why we comment out the java.awt.Color class, and the same reason applies to java.awt.Font in the greenfoot Java source code.

6, study the first scenario of chapter 1: “leaves-and-wombats”, and go over textbook section 1.1 to 1.6;

then study the second scenario of chapter 1: “asteroids1”, and go over textbook section 1.7 to 1.11.

7, install “Eclipse IDE for Java Developers”. As of Jan 2020, the latest version is “**Eclipse 2019-12”.**

This link <https://www.eclipse.org/downloads/packages/release/2019-12/r/eclipse-ide-java-developers> has the download links for all three OS platforms: Windows, Linux, and MacOS.

Please **do not** download and run the Eclipse installer file such as an exe file for Windows or Linux, and instead, use the two links below to download the eclipse zip file for each platform:

**For Windows**, click this link and download the zip file:

<https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2019-12/R/eclipse-java-2019-12-R-win32-x86_64.zip>

**For Linux**, click this link and download the zip file:

<https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2019-12/R/eclipse-java-2019-12-R-linux-gtk-x86_64.tar.gz>

For both Windows and Linux, you need to UNZIP this zip file to a folder such as c:\Eclipse\ for Windows or /home/user/Eclipse for Linux. After unzip, you will find a file name “***eclipse.exe***” in the unzip folder. And that is it! You are done with Eclipse installation. **There is no need to run any installation file**! Don’t use the “download installer” option for Windows and Linux, because it is easier to just download and unzip the zip file.

To make it easier to access the ***eclipse.exe*** without having to go to the extracted folder such as c:\“Eclipse” every time you want to run eclipse, you can create a shortcut for the “eclipse.exe” file and place the shortcut in the desktop in Windows or Linux, or the dashboard in Mac. In Windows “File Explorer”, browse to the folder where “eclipse.exe” file locates, right click this file, select menu item “Send to” 🡪 “Desktop (create shortcut)” , and then a shortcut for Eclipse has been created in the desktop, and now you can run Eclipse by clicking the shortcut.

**For Mac,** click this link:

<https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2019-12/R/eclipse-java-2019-12-R-macosx-cocoa-x86_64.dmg>, then download and run the installation file with file extension dmg.

If you are asked to make a donation in the download webpage, remember that it is totally voluntary, and you don’t need to pay anything to download Eclipse (at least this is true for academic use).

8, Follow document “**CompileAndRunTheFirstJavaProgram.docx**” to compile and run the first Java program using Eclipse. You don’t need to submit anything for the Eclipse project.

9, follow the instructions in file “**homework1.docx**”, and work on homework 1. After you finish homework 1, you need to submit the solution zip files to its Moodle drop box. When coding your homework, please follow all the rules in file “***RulesForIndentAndAlignCode.docx***”. After you click the Moodle drop box for this homework, you will see its due day indicated. The same applies to all other homework assignments in the semester.

10, read file “**LabTutorSchedule.docx**” in Moodle chap 1 folder. Lab tutors are available in the Student Computer Center on the library third floor.

11, if you are UCO or OU transfer students, please check out the following webpage for degree requirements.

For UCO Computer Science transfer: <http://www.occc.edu/catalog/2019-2020/degree-programs/business-and-information-technology/computer-science-computer-science-option-transferring-to-uco-and-colleges-with-similar-patterns-as.html>

For OU transfer Computer Science transfer <http://www.occc.edu/catalog/2019-2020/degree-programs/business-and-information-technology/computer-science-software%20development-transferring-to-ou-and-colleges-with-similar-patterns-as.html>

12, Additional java source code editor, besides the built-in editor in Eclipse IDE and Greenfoot IDE are listed below. They are light-weight software for viewing or editing java source code.

For windows user, you can download and install Notepad++ from this link:

<https://notepad-plus-plus.org/download> , and you may want to download the 64-bit version.

For Mac user, you can download and install BBEdit from this link:

<http://www.barebones.com/products/bbedit/> , it has a trial version for 30 days, and after that it will still remain free. You don’t need to pay anything. Other source code editors for Mac include

<https://macromates.com/download> TextMate

<https://atom.io/> Atom

13, please download file “**Test1StudyGuide.docx**” from Moodle folder “test 1 review lesson”, and start reading this document and answer the questions on it. You need to place the answers right below the questions with a different font color so that answers are distinguished from questions. You need to start preparing for test 1 now, which is the first class of the semester. If you answer all questions in the study guide, you will get 10 points bonus in test 1, even though your answers are not 100% correct.

14, next lesson, we will be working on Moodle folder “chap 2”, please preview its material in Moodle.

15, the core material for this java course is the concept of Object-Oriented Programming (OOP), and I summarize the **four principles** of OOP, **data abstraction, encapsulation, inheritance,** and **polymorphism, as below:**

* **data abstraction**
* **encapsulation**
* **inheritance**
* **polymorphism** with the following context:
* **method overloading**
* **method overriding**
* **parent reference refers to child object**
* **dynamic binding**
* **static binding** (generic programming)

Throughout the semester, we will use examples to explain what the meaning of the above four principles. If you do not understand the above four principles as for now, don’t worry, because we will have the whole semester to study them.

16. this step has a Q & A section:

**Question**: How to be successful in this course?

**Answer**:

First, **READ, READ,** and **READ**. Lots of reading. No shortcut for that. You need to **read** the textbook chapter contents and its source code carefully, then you need to **read** the schedule document of each Moodle lesson carefully, along with any additional WORD documents and source code files in the Moodle lesson folder

Second, for every homework, you need to **read** the homework description file carefully and make sure you understand the requirement in detail before you start working on the homework assignment.

Third, for Greenfoot coding homework, you are required to modify existing greenfoot projects. Then you need to **read** the existing greenfoot source code files that are related to the homework assignment, and understand the existing code, before you start introduce new code to meet the homework requirements.

Fourth, for Eclipse coding homework, you are required to create new Eclipse project and java source files from scratch. Then you need to **read** the sample Java source code file provided in Moodle folder, and then you need to **build the connections from sample Eclipse java source code files to homework requirements**, in order to finish the homework. If there is anything that you don’t know how to code in homework requirement, then you need to ask yourself this question: where is this topic/issue being addressed in the sample Eclipse java files in Moodle folder? Then locate the similar solution in the sample Java files, then build a similar solution in your homework. Emulate the sample source code file provided in Moodle is important!

In summary: **READ, READ** and **READ**! Lots of reading!!! And **build the connections from examples to homework**!

17. this step is very important! **Keep reading!** Last but not least, let’s talk about **how to prepare yourself for this java course**. CS2163 Java course has a prerequisite course CS1143 “Beginning Programming”. There are three different ways to satisfy this course prerequisite:

* take CS1143 and pass with at least C, or
* take a course from other colleges that is equivalent to CS1143 and pass with at least C, or
* take a waiving exam that waives CS1143

No matter which scenario applies to you, you must meet the prerequisite before you enroll in Java. You cannot enroll in “Beginning Programming” and Java at the same semester.

For students who inquire me about what are the topics covered in the waiving exam, here is my reply:

The following topics are covered in the waiving exam that waives CS1143, and these topics are covered in CS1143:

* sequential structure in programming
* conditional structure including if-else statement and switch-case statement
* looping structure including for-loop, while-loop, and do-while loop
* the creation of array; the use of array with the help of loop structure
* the creation and calling of functions (methods), including plugging in suitable parameters for functions, and making use of the return value of functions
* conversion between binary, hexadecimal and decimal number systems
* you can choose your favorite programming language that you want to use in this waiving exam

So, for every student who is now taking Java class, you should also ask yourself this question: **how do I understand the topics listed above**, which are covered in CS1143 “Beginning Programming”?

Regardless how you satisfy the prerequisite of CS2163 Java, you need to go through the topics listed above, and make sure you understand them and how to write code about them, in C or another language that you are familiar with. This Java course assumes that you know the topics listed above very well, and you need to guarantee that this assumption is true. For example, you should already know how to create function, call a function, and make use of the return value of a function. The focus of Java class in the OOP, and in order to study OOP, you have to know how to use function (method) first, **and if you are still struggling with any of the topics above such as functions, or arrays, or loops, or conditional statements, then you may not be ready for this Java course**.

If you have an equivalent course from other college that replaces CS1143 “Beginning Programming”, but you are not familiar with the topics listed above, then **you may consider taking CS1143 “Beginning Programming” first.**

If you took CS1143 “Beginning Programming” at OCCC quite a while ago, and you don’t think you understand the material very well (please refer to the topic listed above), then **you may consider re-taking CS1143 “Beginning Programming” at OCCC** to refreshing your understanding of basic programming skills.

In the end, for whatever other reason, if you don’t think you understand the topic listed above, then you need a remedy, and you need to study the materials in this link:

<https://beginnersbook.com/java-tutorial-for-beginners-with-examples/> , and focus on section: “Tutorial with Java basics”, which has all the topics listed above except methods.

For java methods, you can refer to these links, and learn how to create, call, and make use of methods.

<https://www.programiz.com/java-programming/methods>

<https://www.tutorialspoint.com/java/java_methods.htm>

<http://www.ntu.edu.sg/home/ehchua/programming/java/J2_Basics.html#zz-10>

Attention: the above remedy links needs to be studied outside of class time by each student himself/herself. There is no shortcut to the topics listed above, and you have to satisfy the prerequisite by either taking(re-taking) CS1143 or going through remedy links by yourself.