

# First Year Java Revision / Second Year Preparation

Before entering second year you should be comfortable with the following topics:

1. **Editing and running Java programs using the Eclipse IDE**
2. **Java classes, objects and methods**
3. **Java types: int, char, boolean, double, String**
4. **Instance variables (fields) and local variables**
5. **Parameter passing and return statements**
6. **Conditionals: if statements and nested ifs**
7. **Iteration: for loops, foreach ('enhanced for'), while and iterators**
8. **Collections; ArrayLists, Arrays, Maps, Sets**
9. **Using the Java API**
10. **Inheritance and polymorphism. Interfaces**
11. **Input / output: System.out.println, readLine, Scanner, Text file input/output**
12. **Basic Graphical User Interfaces: frames, textboxes, layout, event handling ...**
13. **Testing and debugging programs**
14. **Exception handling**

## Suggested Reading

Work through the following textbook:

**Head First Java, Sierra and Bates, O'Reilly publishing, 2005. ~ £20 on Amazon.**

**Must cover first 9 chapters.**

Other important chapters: 10, 11, 12, 13, 14 (Text files), 16 (Collections).

**Try to do as many exercises as possible – at least all those highlighted below. Make notes on everything you don't understand and use the Java Workshops to get help.**

**It is estimated that all this will take between 40 and 100 hours, including reading all the chapters and doing the exercises, depending on your existing programming knowledge and experience.**

## Installing Java and Eclipse

To get Java and Eclipse running on your own machine you need to download both **Java** and **Eclipse**.

**Download and install the latest version of Java:**

<https://java.com/en/download/>

Download and install the latest version of **Eclipse IDE for Java Developers**. There is an **installer** that does this for you:

<https://www.eclipse.org/downloads/installer-instructions.php>

(Eclipse is configured to use Java 7. If you wish to change to Java 8 you need to use **Window – Preferences** and set Compiler to **1.8** and also to set the **Installed JREs** to point at your folder/directory that holds your Java 8 JRE e.g. C:\Program Files (x86)\Java\jre1.8.0\_60 )

## Getting Started with Eclipse and Java: Use the Video Lessons

Follow **Lesson 1** of the following **video tutorial**. Keep pausing the tutorial and performing the same actions on your machine (your version of Eclipse may look slightly different from the older video version).

<http://eclipsetutorial.sourceforge.net/totalbeginner.html>

You may find it useful to work through some of the other Lessons, particularly 2, 9, 16, and perhaps 14.

## Working Through Head First Java (HFJ) Book

### Suggested reading order:

Chapters **1-13, 14, 16** and then, if you wish (though not required for second year), the rest of the book: 15, 17, and 18.

Below, for each chapter, is advice on what exercises you should attempt and what programs you should create and run.

**Remember to keep your programs and take notes to ask for help from the demonstrators at the Java Workshop.**

## HFJ Chapter 1

Read through the chapter.

### PROGRAMMING TASK:

**HFJ page 23 Mixed Messages.**

Using Eclipse, create the **class Test** and, in turn, add the **5 different blocks of code**, making sure you work out what the corresponding output should be in advance.

**You can either make a new Eclipse Project for each exercise and add the classes to the src folder or make a new ‘Package’ within the same Project for each exercise and name the packages Exercise1 etc. The Eclipse Tutorial 1 tells you how to create Projects and Packages.**

## HFJ Chapter 2

Read through the chapter.

### PROGRAMMING TASK:

**HFJ page 44 Pool Puzzle.** Using the code snippets and the **two** class skeletons, create and run the code to produce the required output. Can you solve the **Bonus Question** as well?

## HFJ Chapter 3

Read through the chapter.

### PROGRAMMING TASK:

**HFJ page 65 Pool Puzzle.** Using the code snippets and the **Triangle** class skeleton, create and run the code to produce the required output, including the **Bonus Question**.

Can you separate the **Triangle** class into two classes: A simple **Triangle** class and a **TriangleTestDrive** class that only contains the **main method** (see p73 for a similar example)?

### Important Lesson:

**At the moment, the main method is accessing the Triangle fields directly e.g. “t5.area”. This is poor practice – all data should be accessed through methods. Later chapters will not do this. You shouldn’t do this either.**

## HFJ Chapter 4

Read the chapter:

**Now we are getting to the important material! Notice the use of private when defining instance variables (fields) and public (or protected or private) when defining methods.**

### PROGRAMMING TASK:

**HFJ page 91, yep it’s Pool Puzzle time again!** At first sight this might look a bit daunting but using the snippets severely limits what can be placed on each blank line – also Eclipse will often warn you if you put the snippet in the wrong place.

Again, try to separate the main method class, say **PuzzleDriver**, from the **Puzzle** class.

### Additional Challenge:

The solution is still accessing the **ivar** instance variable (field) directly. This should really be done through another method such as **setiVar(int i)**. Can you add this to the **Puzzle** class and change the main method to use this method rather than **obs[x].ivar**?

## HFJ Chapter 5

### PROGRAMMING TASK:

**Now a serious bit of programming.**

**Work carefully through the whole chapter using Eclipse to build up and run the ‘Dot Com Game’.**

Start with ‘Simple Dot Com Game’ – see if you can get the **SimpleDotComTestDrive** and **SimpleDotCom** classes written yourself by following **pages 95-105** before you get to **page 106**.

Now try the exercise on **page 107** – the actual **game class**. You can then use the code on **page 110**.

Now you need a third helper/utility class to get some input for you. Just copy the third class, **GameHelper**, from the **HFJ book page 112** into Eclipse.

**With the aid of the HFJ book, try to understand all the code** (except the **GameHelper** class).

**The Bug!** Can you fix the bug highlighted on **page 113**? Don't worry if you can't, we'll fix it in the next exercise.

Keep working through the chapter ... **there's lots of good stuff there: enhanced for (foreach), for v. while loops, converting String to an int, casting primitives.**

Make sure to try all the exercises – run them in Eclipse if you wish.

## HFJ Chapter 6

Read through the chapter as usual. Here you'll find the solution to the bug in Chapter 5!

**Important stuff here on ArrayLists, and comparing them with Arrays.**

### **PROGRAMMING TASK:**

**Using ArrayLists, as per page 139, correct the bug in the Chapter 5 'Dot Com Game'.**

**Work through the rest of the chapter – again, there's lots of important material on the 'Java API' (libraries) here.**

[You don't need to build the 'final version of the DotCom class' but you may find it useful to do so.]

## HFJ Chapter 7

**Read through the chapter – there's tons of important material on Inheritance and Polymorphism!**

**Work through the chapter doing the paper exercises – it's all important.**

**PROGRAMMING TASK** (fairly straightforward):

**HFJ page 193.** Create the 4 classes: **MonsterTestDrive**, **Monster**, **Vampire** and **Dragon**. Complete **Monster** and **Vampire** to get the desired output.

## HFJ Chapter 8

**Read the chapter doing the paper exercises as you go. These paper exercises are really important.**

**Key material: Abstract Classes, Abstract Methods, Interfaces and the Object class. See the BULLET POINTS on page 229.**

There are no programming exercises for this chapter.

## HFJ Chapter 9

Read the chapter.

**Key material: object creation and memory allocation, constructors, primitive types, object references, superclass constructors ( `super()` ), scope, object deletion, instance variables, local variables.**

**Some very useful exercises here: attempt those on pages 257, 266 and 267.**

[Could also look at the Five-Minute Mystery on page 268.]

## HFJ Chapter 10

Reading the chapter.

**Key material:** static methods, static variables, use of 'final', Math methods, wrapping primitives, autoboxing, converting between numbers and Strings.

Attempt the exercise on page 285.

### PROGRAMMING TASK:

**HFJ page 291.** Create the class **TestBox** using Eclipse and see what happens. Can you work out why? Try changing the assignment to be **i = j** rather than **j = i**

Can now skim through formatting numbers and dates, static imports and the remaining exercises.

## HFJ Chapter 11

The next three chapters build up a fairly sophisticated set of Music Machine examples.

**You have a choice now, if you have time and enthusiasm it would be very useful (and entertaining!) to try to build the examples as you work through the chapters. Otherwise, you should make sure you understand the following topics:**

**Exceptions pages 318-327. Try the exercise HFJ page 328.**

**Ducking exceptions pages 335-337.**

**HFJ Exercise page 348: True or False.**

As the book says, you may skip page 339 until the end of Chapter 11 – the Code Kitchen stuff.

## HFJ Chapter 12

The Graphics material in Chapters 12 and 13 is not critical for second year. However, there is still a lot of useful material in here, and you will need to create Graphical User Interfaces in later years, so it is worth reading through and trying to understand the following key ideas:

**Frames and widgets (Components), Getting user events, GUI layout, Inner classes to handle events**

### PROGRAMMING TASK:

**Copy the SimpleGui1 code from HFJ page 355 and get it running within Eclipse.**

(Notice it doesn't react to Button Events!)

### PROGRAMMING TASK:

**Add the action handling code given on page 360.** Now it should react to the Button being 'clicked'!

[ Optional Programming Task:

If you wish, you can play around with some of the Graphics programs on pages 364 to 385, and then onto the Code Kitchen again. ]

**Finally, attempt the HFJ 'Who am I?' Exercise page 394.**

## HFJ Chapter 13

It is worth having a quick read through the key ideas associate with **Layout Managers** and the Components `JTextField`, `JTextArea`, `JCheckBox` and `JList`.

**Attempt HFJ Exercise pages 424-425 ‘Which code goes with which layout?’**

[ Optional Programming Task:

The full Beat Box example finishes off here in the Code Kitchen. ]

## HFJ Chapter 14

**The key material in this chapter is on Text Files pages 447-459.**

(Try not to be put off by the large `QuizCardBuilder` example, just focus on reading from files.)

You can quickly skim through the initial serialisation and deserialisation material pages 429- 446.

You can skip pages 460-469.

## HFJ Chapter 15

This chapter is about connecting to programs on other machines and running multiple threads – more advanced topics for later years. You can safely skip over this chapter.

## HFJ Chapter 16

Data structures, sorting, and generics are key` topics for **second year**, so again you can safely skim through this chapter.

It is however useful to know the key properties of the main **Collection** classes covered in the first year Java programming class: **ArrayLists, Arrays, HashMap, HashSet.**

## HFJ Chapter 17

Releasing your code is background material that you don’t need to worry about at this stage. It is useful to know how to create an executable jar file, but you’ll be shown this in later years. Eclipse makes it very easy to create an executable jar.

## HFJ Chapter 18

Distributed computing is a topic for later years and can be skipped.