Download and Set Up (Cooperative Runtime)

HJlib provides support for multiple runtimes. In this article, we will discuss how to set up the JDK11 compatible cooperative runtime. Instructions on the download and set up of the thread-blocking runtime is available in another article.

Here are the steps for running HJlib on Java 11 using an IDE:

Step 1: Java 11 Installation

You will also need a Java 11 installation on your machine and have your JAVA_HOME and PATH point to the new installation. Java 11 can be downloaded and installed from the <u>Oracle website</u>.

For e.g., I have the following on my Mac machine's .bash_profile:

JAVA11_HOME=/Library/Java/JavaVirtualMachines/jdk11.0.13.jdk/Contents/Home export JAVA_HOME=\${JAVA11_HOME} export PATH=\$JAVA_HOME/bin:\$PATH

On Windows the environment variables have to be set up differently, refer to this stackoverflow question to see how it can be done.

Step 2: HJlib JAR File

Download the JDK11 compatible HJlib jar file and save it to a local directory.

Step 3: IDE like IntelliJ or Eclipse

HJlib is a simple Java library (jar file) and can be used in any Java project. As such a simple text editor can be used to write programs that use HJlib.

However, we recommend using an IDE like IntelliJ to do the Java development using HJlib in the labs and assignments.

A free version of IntelliJ (Community Edition) can be downloaded and installed from the Jetbrains website.

Step 4: Your first project

We show how to set up a simple Maven project with a HJlib HelloWorld program in IntelliJ. Similar steps can be followed for other IDEs like Eclipse.

First, download the helloworld.zip file and unzip it into a directory of your choice (e.g. /Users/shamsimam/projects/comp322-s2015-projects /helloworld).

Next, import this project into IntelliJ using the File -> Import Project menu option.



This should show the following series of popups:

(e O O Select File or Directory to Import	
o ir	Select directory with existing sources , Eclipse project (.project) or classpath (.classpath) file, Maven project file (pom.xml), Gradle build script (*.gradle).	
l		path
L	/Users/shamsimam/projects/comp322-s2015-projects/helloworld	
ł.	comp322-s2015-projects	
	🔻 🛅 helloworld	
)- 1.	src pom.xml README	
	Drag and drop a file into the space above to quickly locate it in the tree. Cancel OK	

00	Import Project
Create project from existing sources	
Import project from external model	
Editore	
 Conse Gradle 	
m Maven	
(?) Cancel	Previous

00	Import Project
Root directory //projects/	comp322-s2015-projects/helloworld
Search for projects recu	irsively
Project format: .idea (dir	rectory based) 🗘
Keep project files in:	
Import Maven projects	automatically
🗹 Create IntelliJ IDEA mod	lules for aggregator projects (with 'pom' packaging)
Create module groups f	or multi-module Maven projects
☑ Keep source and test fo	lders on reimport
Exclude build directory	(%PROJECT_ROOT%/target)
🗹 Use Maven output direc	tories
Generated sources folders:	Detect automatically \$
Phase to be used for folders	update: process-resources 🗘
IDEA needs to execute one Note that all test-* phases	of the listed phases in order to discover all source folders that are configured via Maven plugins. firstly generate and compile production sources.
Automatically download:	Sources Documentation
Dependency types: jar, te	st-jar, maven-plugin, ejb, ejb-client, jboss-har, jboss-sar, war, ear, bundle
Comma	separated list of dependency types that should be imported
	Environment settings
? Cancel	Previous

00	Import Project	
Select profiles:		
helloworld		
(?) Cancel		Previous Next

● ○ 0	Import Project
Select Maven projects to import	
du.rice.comp322:lab1:1.0-SNAPSHOT	
	Select all Unselect all
Open Project Structure after import	
? Cancel	Previous Next

00	Import Project			
Please select project SDK. This SDK will be used by default by all project modules.				
+ -	Name: 1.8			
1.7				
1.8	IDK home path: rv/lava/lavaVirtualMachines/idk1.8.0_05.idk/Contents/Home			
🖷 Android API 19 Platf				
DEA IC-135.909	Classpath Sourcepath Annotations Documentation Paths			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/lib/ant-javafx.jar			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/lib/dt.jar			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/lib/javafx-mx.jar			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/lib/jconsole.jar			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/lib/sa-jdi.jar			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/lib/tools.jar			
	Library/Java/JavaVirtualMachines/Jdk1.8.0_05.jdk/Contents/Home/Jre/lib/charsets.jar			
	Library/Java/JavaVirtualMachines/Jdk1.8.0_05.jdk/Contents/Home/Jre/lib/depioy.jar			
	/Library/Java/Java/irtualMachines/Juk1.8.0_05.juk/Contents/Home/jre/lib/iavaws jar			
2	// ibrary/Java/JavaVirtualMachines/Jdk1.8.0_05.jdk/Contents/Home/ire/lib/javaws.jar			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/jre/lib/jfr.jar			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/jre/lib/jfxswt.jar			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/jre/lib/jsse.jar			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/jre/lib/management-ag			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/jre/lib/plugin.jar			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/jre/lib/resources.jar			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/jre/lib/rt.jar			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/jre/lib/ext/cldrdata.jar			
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/jre/lib/ext/dnsns.jar			
	Library/Java/JavaVirtualMachines/Jdk1.8.0_05.jdk/Contents/Home/Jre/lib/ext/Jtxrt.jar			
	Library/Java/Java/VirtualMachines/Jok1.8.0_05.jok/Contents/Home/Jre/lib/ext/localedata.J			
	/Library/Java/JavaVirtualMachines/Juk1.8.0_05.juk/Contents/Home/ire/lib/ext/super_jar			
	+ -			
? Cancel	Previous			

00	Import Project			
Please enter a name to create a new IntelliJ IDEA project.				
Project name:	helloworld			
Project file location:	~/projects/comp322-s2015-projects/helloworld			
? Cancel	Previous	nish		

First enable the project view, then right-clicking on the root folder to change the module settings.

View	Navigate	Code	Analyze	Refactor	Build	Run	Tools	V
Тоо	Windows		•	🖲 Projec	t	ж	1	
Quio	k Definition	n	☆業Ⅰ	★ Favori	tes	#	2	
Quic	w Siblings k Documer	itation	^J	Mebug	9	90 90	5	
Ope	n Module S	ettings	F4	V Struct	ure	¥	7	
Rece Rece Rece	ent Files ently Chang ent Changes	ed Files	#E ዕ#E ጊዕፓ	ि Chang असे Ant Bu बि Applic	jes Jild ation Se	#	9	
Quio	k Switch Sc	heme	<u>^`</u>	🖀 Comm 🖬 Desigi	n ander ner			
Too Too ✓ Stat ✓ Navi	lbar I Buttons us Bar igation <mark>B</mark> ar			Event Maver Palette Palette	Log Project	5		
Ente Ente	er Presentati er Full Scree	ion Mode n	° ^₩F	I Versio	na Contr	ol	F12	

he	lloworld		
🗊 Pr	oject	•	C
-	helloworl	lah11 (~/nrojects/comn322-s2	015-projects/
►	🗖 .idea	New	•
►	src 🛛	Add Framework Support	
	.gitigr	🖌 Cut	жx
	🗾 lab1.i	Copy	жc
	m pom.:	Copy Path	企業C
	READ	Conv Reference	大 介留C
	External	n Paste	£100€ ₩V
		Find Usages	℃F7
		Find in Path	^
		Replace in Path	<u>^ </u>
		Analyze	•
		Refactor	•
		Clean Python Compiled F	iles
		Add to Favorites	•
		Show Image Thumbnails	ት¥T
		Reformat Code	\C₩L
	_	Optimize Imports	7.80
		Remove Module	\boxtimes
		Maka Madula lah 1	
	_	Make Module Tabi	A 99 F 0
	_		Υ#F9
	_	Create All Tests	0.0510
	_	Run All Tests	~~~F10
	_	Debug 'All Tests'	~&F9
		Run All Tests with Cover	rage
		Local History	►
		Synchronize 'helloworld'	
		Reveal in Finder	
		Reveal III I IIIdei	
		Compare Directory with	%D
		Open Module Settings	F4
		Move Module to Group	►
		Mark Directory As	►
		Create Cist	
		m Mayon	•
		Maven	P*

We will be using Java 8 lambdas while developing with HJlib, so we need to ensure the language level settings in the project:

00	Project Structure
⋡ ⇒	Project name:
Project Settings	helloworld
Project	
Modules	Project SDK:
Libraries	A module specific SDK can be configured for each of the modules as required
Facets	□ 1.8 (java version "1.8.0_05")
Artifacts	
Platform Settings	This language level is default for all project modules.
SDKs	A module specific language level can be configured for each of the modules as required
Global Libraries	8 - Lambdas, type annotations etc.
Problems	Project compiler output: This path is used to store all project compilation results. A directory corresponding to each module is created under this path. This directory contains two subdirectories: Production and Test for production code and test sources, respectively. A module specific compiler output path can be configured for each of the modules as required
	/Users/shamsimam/projects/comp322-s2015-projects/helloworld/out

Notice, the library dependencies have already been resolved thanks to maven:

00	Project Structure
 Project Settings Project Modules Ibraries Facets Artifacts Problems 	Name: Maven: edu.rice:hjlib-cooperative:0.1.4-SNAPSHOT //Sers/shamsimam/.m2/repository/edu/rice/hjlib-cooperative/0.1.4-SNAPSHOT/hjlib-cooperative-0.1.4-SNAPSHOT.jar © Sources @/Users/shamsimam/.m2/repository/edu/rice/hjlib-cooperative/0.1.4-SNAPSHOT/hjlib-cooperative-0.1.4-SNAPSHOT-sources.jar © JavaDocs @/Users/shamsimam/.m2/repository/edu/rice/hjlib-cooperative/0.1.4-SNAPSHOT/hjlib-cooperative-0.1.4-SNAPSHOT-javadoc.jar

You will need to note the path of the HJlib jar file as we will need it to be able to run the HelloWorld.java program from IntelliJ. On my machine it is at /Users/shamsimam/.m2/repository/edu/rice/hjlib-cooperative/0.1.4-SNAPSHOT/hjlib-cooperative-0.1.4-SNAPSHOT.jar

Open the HelloWorld.java file in the editor and attempt to run it by right clicking on it and choosing the Run option:



Since we are running HelloWorld. java without configuring the javaagent option, it is likely to intermittently fail with an error as follows: java.lang. Error: Calling function not instrumented

44					
Run 🖶 Hel	un 🕾 HelloWorld				
	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/bin/java /Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/bin/java java.long.Error: Calling function not instrumented at edu.rice.hj.continuation.Coroutine.yield(<u>Coroutine.java:57</u>) at edu.rice.hj.runtime.baseruntime.DataDrivenControl.pauseOnDdc(<u>DataDrivenControl.java:28</u>) at edu.rice.hj.runtime.baseruntime.DataDrivenControl.pauseOnDdc(<u>DataDrivenControl.java:27</u>) at edu.rice.hj.runtime.baseruntime.Finishtate.waitForFinish(<u>FinishState.java:134</u>) at edu.rice.hj.runtime.baseruntime.Finishtate.waitForFinish(<u>FinishState.java:134</u>) at edu.rice.hj.runtime.baseruntime.KaseRuntime.java:210 at edu.rice.hj.runtime.baseruntime.KaseRuntime.java:210 at edu.rice.hj.runtime.baseruntime.Finishtate.waitS.a.excuteBody(<u>HabaneroActivity,java:351</u>) at edu.rice.hj.runtime.baseruntime.Finishtate.waitS.a.excuteBody(<u>HabaneroActivity,java:350</u>) at edu.rice.hj.runtime.baseruntime.SuspendableActivity.excuteBody(<u>HabaneroActivity,java:38</u>) at edu.rice.hj.runtime.baseruntime.SuspendableActivity.giva:350 at edu.rice.hj.continuation.Coroutine.run(<u>Coroutine.java:155</u>) at edu.rice.hj.continuation.Coroutine.run(<u>Coroutine.java:155</u>) at edu.rice.hj.runtime.baseruntime.SuspendableActivity.processCoroutine(<u>SuspendableActivity,java:77</u>) at edu.rice.hj.runtime.baseruntime.SuspendableActivity.java:155 at edu.rice.hj.runtime.baseruntime.SuspendableActivity.java:155 at edu.rice.hj.runtime.baseruntime.SuspendableActivity.java:155 at edu.rice.hj.runtime.baseruntime.SuspendableActivity.java:155 at edu.rice.hj.runtime.baseruntime.SuspendableActivity.java:155 at edu.rice.hj.runtime.baseruntime.SuspendableActivity.java:155 at edu.rice.hj.runtime.baseruntime.SuspendableActivity.java:155 at edu.rice.hj.runtime.baseruntime.SuspendableActivity.java:25 at edu.rice.hj.runtime.baseruntime.SuspendableActivity.java:25 at edu.rice.hj.runtime.baseruntime.SuspendableActivity.java:25 at java.util.concurrent.ForkJoinTaskSRunnableExecuteAction.exec(ForkJ				

Or the following error:

Error occurred during initialization of VM agent library failed to init: instrument

To fix this error, we need to correct set up the javagent JVM option before attempting to run HelloWorld.java.

Java agents provide services that allow Java programming language agents to instrument programs running on the JVM. To run HJlib programs under the cooperative runtime we need to configure the JVM to use agents available in the HJlib jar. An agent is started by adding this option to the command-line:

-javaagent: jarpath[=options]

We need to configure this in IntelliJ to be able to successfully run HJlib programs. Below is an image of what the configuration looks like after editing the run configuration:

	M.
👫 🔚 HelloWorld 👻	📡 🖄 🧏 🕵 🔁 🔂 📾 📿
📑 Edit Configura	ations
Save 'HelloWo	orld' Configuration
es	
000	Run/Debug Configurations
+ − 🖺 🖶 🍄 🔺 🕆 🖿 ↓² Name: Helle	oWorld Share Single instance only
Application HelloWorld	Configuration Code Coverage Logs
▶ Sefaults Main class	s: edu.rice.comp322.HelloWorld
V/M ention	ri invester //kom/instructure/m2/macing//du/dig/bilk_constration/0.1.4_SNARUOT/bilk_constration_0.1.4_SNARUOT is:
Program a	
Working d	iguments.
Emironme	rectory. (Jose's sharisinan) projects/comp522-s2013-projects/menowonu
Environme	m variadnes.
Use classp	ath of module: 📑 lab1 🗘
Use al	ternative JRE:
Enable	e capturing form snapshots
Before launch: Make	
+ [®] Make	
5	
ne Show this	s page
pe	
ri (?)	Cancel Apply OK

Now we can run the program by clicking on the green play button. Running the file on my machine produces the following output for example:

		40 3
Run	📄 Hel	loWorld
	+	/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/bin/java
	÷	First
Ш	<u>97</u>	Third Fourth
O		Second
-8		