

Programming Environment

Outline

① Programming Environment

② Working with the *j--* Compiler

Programming Environment

Programming Environment

Linux, Mac, or Windows operating system configured with the software needed for the course



Programming Environment

Linux, Mac, or Windows operating system configured with the software needed for the course

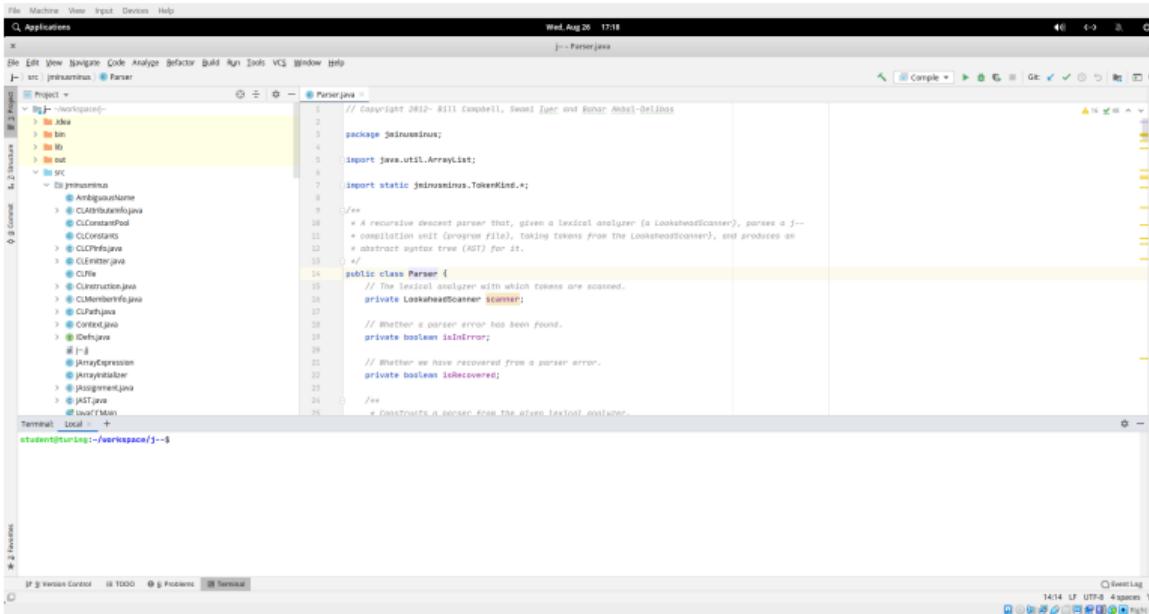


Tools we will use: web browser 🌐, IntelliJ IDE 🖥, terminal ☈, and file manager 📁

Programming Environment

Programming Environment

Use IntelliJ IDE  to create/edit/debug/run Java programs



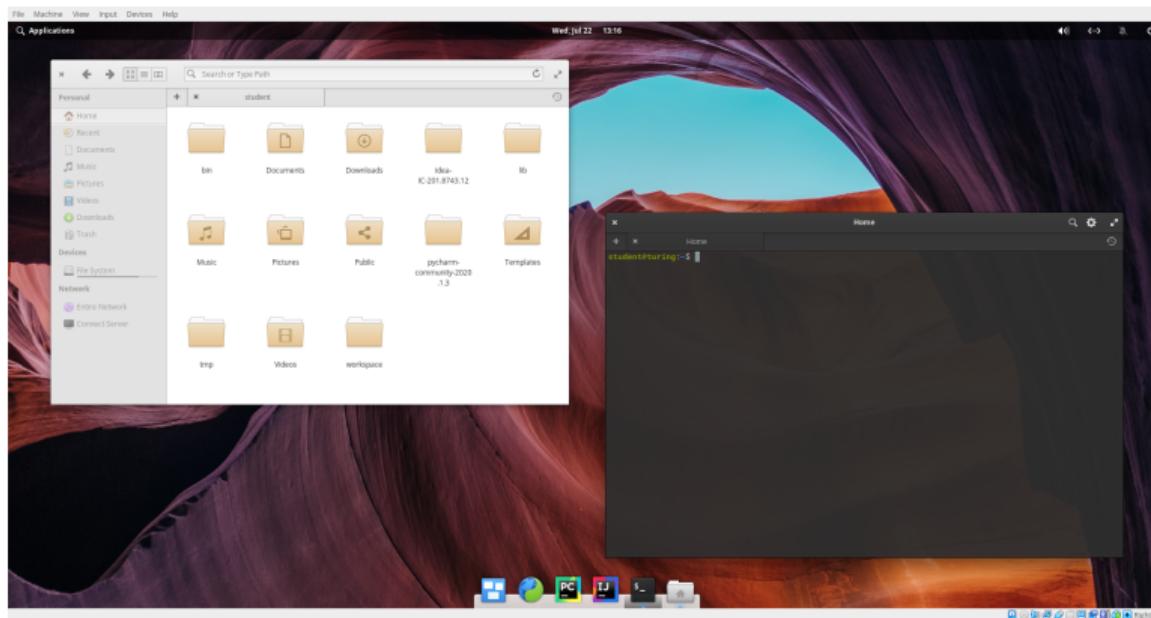
The screenshot shows the IntelliJ IDEA interface with the following details:

- File Menu:** File, Machine, View, Input, Devices, Help.
- Toolbar:** Applications, Wed Aug 26 17:18, j -> Parser.java.
- Project Tool Window:** Shows the project structure with packages like `jp`, `jp.jminimis`, and `Parser`.
- Code Editor:** Displays the `Parser.java` file content. The code is a recursive descent parser for Java-like syntax, utilizing a lexical analyzer (`LookaheadScanner`) and producing an abstract syntax tree (`AST`). It includes imports for `java.util.ArrayList` and `TokensKind`, and defines classes like `Parser` and `Scanner`.
- Terminal:** Shows the command `student@turing:~/workspace/j->`.
- Bottom Status Bar:** Event Log, 14/14 LF, UTF-8, 4 spaces, Right Click.

Programming Environment

Programming Environment

Use terminal  and file manager  to navigate the file system



Working with the j-- Compiler

Working with the j-- Compiler

Download <https://www.cs.umb.edu/j--/j--.zip> under `~/workspace` (`~` is shorthand for the user's home directory)

Working with the j-- Compiler

Download <https://www.cs.umb.edu/j--/j--.zip> under `~/workspace` (`~` is shorthand for the user's home directory)

Launch terminal

```
>_ ~  
$ _
```

Working with the j-- Compiler

Download <https://www.cs.umb.edu/j--/j--.zip> under `~/workspace` (`~` is shorthand for the user's home directory)

Launch terminal

```
>_ ~  
$ _
```

Change directory to `~/workspace`

```
>_ ~  
$ cd ~/workspace
```

Working with the j-- Compiler

Download <https://www.cs.umb.edu/j--/j--.zip> under `~/workspace` (`~` is shorthand for the user's home directory)

Launch terminal

```
>_ ~  
$ _
```

Change directory to `~/workspace`

```
>_ ~  
$ cd ~/workspace
```

Extract the downloaded zip file

```
>_ ~/workspace  
$ unzip j--.zip
```

Working with the j-- Compiler

Working with the j-- Compiler

List current directory

```
>_ ~/workspace  
$ ls  
j--  j--.zip
```

Working with the j-- Compiler

List current directory

```
>_ ~/workspace  
$ ls  
j--  j--.zip
```

Remove the zip file

```
>_ ~/workspace  
$ rm j--.zip
```

Working with the j-- Compiler

List current directory

```
>_ ~/workspace  
$ ls  
j-- j--.zip
```

Remove the zip file

```
>_ ~/workspace  
$ rm j--.zip
```

Exit the terminal

```
>_ ~/workspace  
$ exit
```

Working with the j-- Compiler

List current directory

```
>_ ~/workspace  
$ ls  
j--  j--.zip
```

Remove the zip file

```
>_ ~/workspace  
$ rm j--.zip
```

Exit the terminal

```
>_ ~/workspace  
$ exit
```

We refer to parent directory of j--, which is ~/workspace in our case, as \$j

Working with the j-- Compiler

Working with the j-- Compiler

Open \$j/j-- in IntelliJ to work on the programs

Working with the j-- Compiler

Open \$j/j-- in IntelliJ to work on the programs

Compile the *j*-- compiler on the IntelliJ terminal

```
>_ ~/workspace/j--
```

```
$ ant
```

Working with the j-- Compiler

Open \$j/j-- in IntelliJ to work on the programs

Compile the *j*-- compiler on the IntelliJ terminal

```
>_ ~/workspace/j--  
$ ant
```

Compile the *j*-- program \$j/j--/tests/jvm/HelloWorld.java for JVM on the IntelliJ terminal

```
>_ ~/workspace/j--  
$ bash ./bin/j-- tests/jvm/HelloWorld.java
```

Working with the j-- Compiler

Open \$j/j-- in IntelliJ to work on the programs

Compile the *j*-- compiler on the IntelliJ terminal

```
>_ ~/workspace/j--  
$ ant
```

Compile the *j*-- program \$j/j--/tests/jvm/HelloWorld.java for JVM on the IntelliJ terminal

```
>_ ~/workspace/j--  
$ bash ./bin/j-- tests/jvm/HelloWorld.java
```

Run the JVM program HelloWorld.class on the IntelliJ terminal

```
>_ ~/workspace/j--  
$ java HelloWorld
```

Working with the j-- Compiler

Working with the j-- Compiler

Disassemble `HelloWorld.class` on the IntelliJ terminal

```
> ~/workspace/j--  
$ javap -p -v HelloWorld
```

Working with the j-- Compiler

Disassemble `HelloWorld.class` on the IntelliJ terminal

```
>_ ~/workspace/j--  
$ javap -p -v HelloWorld
```

Compile the *j--* program `$j/j--/tests/spim/HelloWorld.java` for MIPS on the IntelliJ terminal

```
>_ ~/workspace/j--  
$ bash ./bin/j-- -s naive tests/spim/HelloWorld.java
```

Working with the j-- Compiler

Disassemble `HelloWorld.class` on the IntelliJ terminal

```
>_ ~/workspace/j--  
$ javap -p -v HelloWorld
```

Compile the *j--* program `$j/j--/tests/spim/HelloWorld.java` for MIPS on the IntelliJ terminal

```
>_ ~/workspace/j--  
$ bash ./bin/j-- -s naive tests/spim/HelloWorld.java
```

Run the MIPS program `HelloWorld.s` on the IntelliJ terminal

```
>_ ~/workspace/j--  
$ spim -f HelloWorld.s
```

Working with the j-- Compiler

Disassemble `HelloWorld.class` on the IntelliJ terminal

```
>_ ~/workspace/j--  
$ javap -p -v HelloWorld
```

Compile the *j--* program `$j/j--/tests/spim/HelloWorld.java` for MIPS on the IntelliJ terminal

```
>_ ~/workspace/j--  
$ bash ./bin/j-- -s naive tests/spim/HelloWorld.java
```

Run the MIPS program `HelloWorld.s` on the IntelliJ terminal

```
>_ ~/workspace/j--  
$ spim -f HelloWorld.s
```

Use the web browser  to sign on to Gradescope and upload files