Apache ANT

CS3310, Language Translators

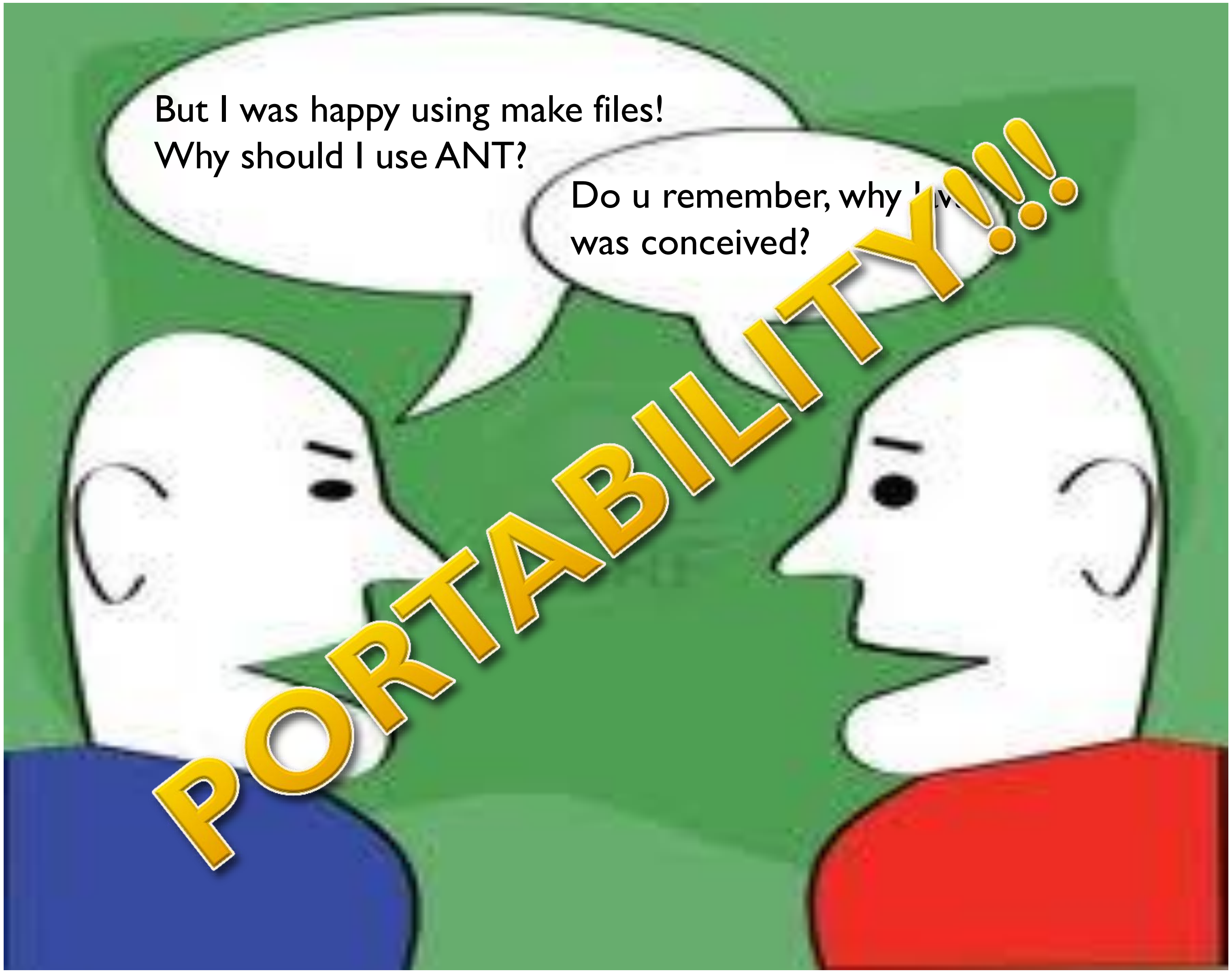
By **Manas Thakur**

ANT???

What is Apache ANT?

- “Another Neat Tool”
- Automates the build process
- Written for Java
- Open-source
- Uses XML

<APACHE ANT>


But I was happy using make files!
Why should I use ANT?
Do u remember, why [illegible] was conceived?
PORTABILITY!!!

Using ANT

- Download the OS-specific contents from ant.apache.org, and setup the PATH (complete details are given there)

- Create a build file(contents are explained next)
  - *build.xml* is the default name

- Run the command ant
  - If your build file is *abc.xml*
    - Run the command ant –f abc.xml

Hello, World!

build.xml

```
<project default="hello">
    <target name="hello">
        <echo message="Hello, World!"/>
    </target>
</project>
```

```
$ ant
```

```
Buildfile: build.xml

hello: [echo] Hello, World

BUILD SUCCESSFUL
```

Multiple Targets

```
<project default="hello">
        <target name="hello">
                <echo message="Hello, World"/>
        </target>
        <target name="goodbye">
                <echo message="Goodbye, Cruel World"/>
        </target>
</project>
```

```
$ ant goodbye
```

```
Buildfile: multitargets.xml

goodbye: [echo]  Goodbye, Cruel World

BUILD SUCCESSFUL
```

Specifying Dependencies

```
<project default="hello">
          <target name="hello">
                    <echo message="Hello, World"/>
           </target>
           <target name="goodbye">
                    <echo message="Goodbye, Cruel World"/>
           </target>
          <target name="all" depends="hello, goodbye" />
</project>
```

```
$ ant all
```

```
Buildfile: build.xml
hello: [echo] Hello, World
goodbye: [echo] Goodbye, Cruel World
all:
BUILD SUCCESSFUL
```

Compile, create jar, and execute a Java source: all in one go!

Hello.java

```
public class Hello {
        public static void main(String[] args) {
                System.out.println(“Hello, World!”);
        }
}
```

- Suppose this file is in current directory (represented by “.”)

Compile, create jar, and execute a Java source: all in one go!

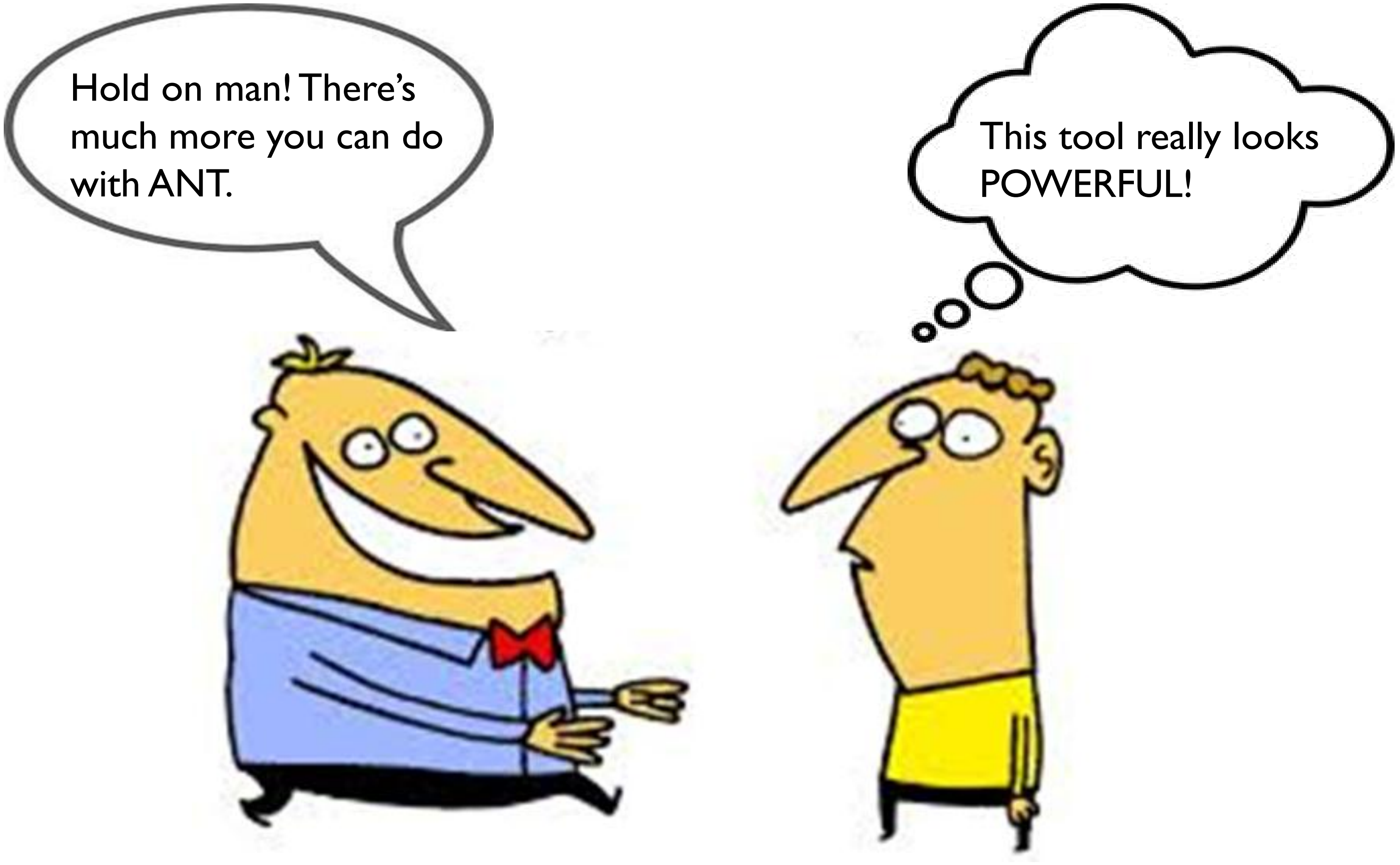
hello.xml

```
<project default="compile">
        <target name="compile">
                <javac srcdir="." />
        </target>
        <target name="jar" depends="compile">
        <jar destfile="Hello.jar" basedir="." includes="**/*.class" />
        </target>
        <target name="run" depends="jar">
                <java classname="Hello" fork="true">
                <classpath path="Hello.jar" />
        </java>
        </target>
</project>
```

Compile, create jar, and execute a Java source: all in one go!

```
$ ant –f hello.xml run

compile:

jar:

run: [java] Hello World

BUILD SUCCESSFUL
```

So, that's it???


Hold on man! There's much more you can do with ANT.
This tool really looks POWERFUL!

Using ANT Properties

```
<project default="all">
        <property name="obj-dir" location="obj" />
        <property name="lib-dir" location="lib" />
        <target name="init">
                <mkdir dir="${obj-dir}" />
                <mkdir dir="${lib-dir}" />
        </target>
        <target name="clean-init">
                <delete dir="${obj-dir}" />
                <delete dir="${lib-dir}" />
        </target>
        <target name="all" depends="init"/>
        <target name="clean" depends="clean-init"/>
</project>
```

Using ANT Properties

- 4 targets:
  - **init**: to build the work directory structure
  - **clean-init**: to remove the work directory structure
  - **all**: the build target that depends on init
  - **clean**: to clean target that depends on clean-init

- 2 properties:
  - **obj-dir**: the root directory for our .class files
  - **lib-dir**: the root directory for our .jar files

Final Code Demonstration in Eclipse (plus a surprise!!)

<APACHE ANT>

+

What a Combination!