



Java Programming

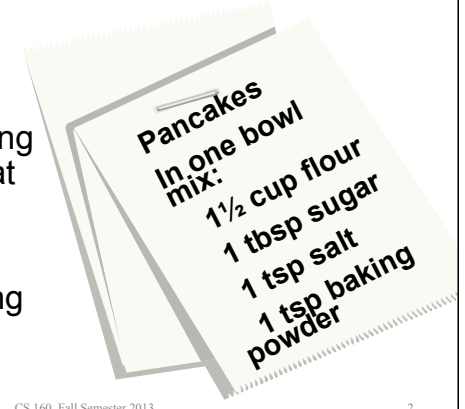
TOPICS

- Computer Programs
- Using Eclipse
- Hello World Program
- Program Components



What is a program?

- Definition: a sequence of instructions telling a computer what to do
- Analogy: cooking recipes



Programs as Recipes

- Recipes specify
 1. What ingredients to use
 2. What to do with them
 - Sometimes conditional: “bake until golden brown”, “salt to taste”
- Programs specify
 1. What information (data) to use
 2. What operations to apply
 - Sometimes conditional: “while file is not empty”, “for every element in array”



Programs ≠ Recipes

- At some point the analogy breaks down:
- Recipes are read by people
 - People can make inferences.
 - If something goes wrong, people react.
 - People sometimes make mistakes.
 - Programs are read by machines
 - Machines do exactly what they are told!
 - No matter how badly things go wrong.
 - But, they never make mistakes or get tired.



Definitions

- **program**
 - a set of directions telling a computer exactly what to do
- **programming languages**
 - precise languages for specifying sequences of directions to a computer
 - unlike English -- no ambiguities! no nuances!
- **algorithm**
 - a sequence of steps to be followed to solve a problem
 - independent of any programming language

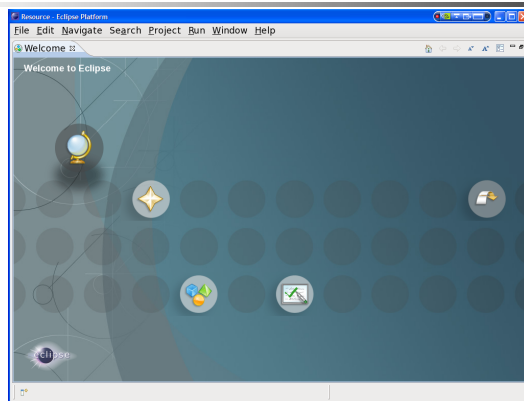


Object-oriented programming

- Java programs use objects and methods
 - An 'object' is a collection of data you treat as a unit
 - Objects have 'methods' – actions that apply to the object
- A Java cooking program might have a line like 'egg.scramble()' to scramble an egg...



What does eclipse look like?

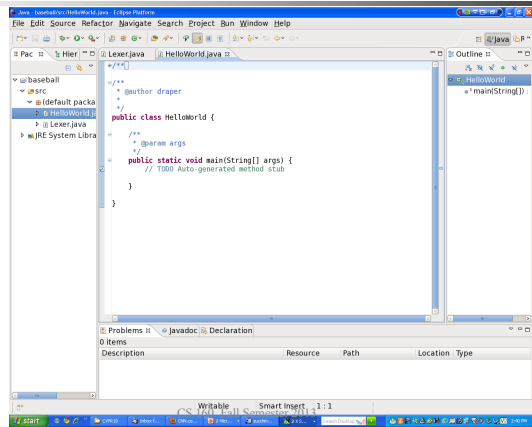


Using Eclipse

- OK, the intro screen isn't too helpful
 - Feel free to hit the 'X' by welcome
 - That will kill the intro screen
- Create a new project using the menus
 - File -> New -> Project
 - Select a Java project
 - Give it a name/directory
- Then create a new class
 - File -> New -> Class
 - It will offer to make stubs, etc.
 - Let it make a main stub for you...



HelloWorld via Eclipse



9



Hello World Program

```
// HelloWorld
// Author: Chris Wilcox
// Date: 1/15/2013
// Class: CS160
// Email: wilcox@cs.colostate.edu

import java.lang.*;

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

CS 160, Fall Semester 2013

10



Import directive

- Tells Java what other classes or packages (collections of classes) to use.
- `java.lang.*` includes the class **System**, always assumed by compiler.
- Must be above the class declaration, at the top of the program.
- Eclipse will automatically create import directives as needed.

CS 160, Fall Semester 2013

11



Comments

- Any line beginning with `//` or any text between `/*` and `*/`.
- Describes the behavior and attributes of a program.
- For humans only – Java compiler ignores all comments when generating code.
- *But really important for humans!*

CS 160, Fall Semester 2013

12



Class Heading

- Creates a new class that can contain code (methods) and data:
 - this one is called 'HelloWorld'
- Every Java file begins with a class heading, which is always public.
- Class name must match file name
 - this file is called HelloWorld.java
- Only one class per file is allowed



Curly brackets { and }

- Used to group statements together into larger units
- Everything within the outer brackets is part of the class **HelloWorld**
- Everything within the inner brackets is part of the method **main**



Class Body

- Everything between { and } of the class header is the class body.
- The class body defines the data in the object and its methods.
- This class has no data, and only one method, which is the program starting point.



Main Method

- The main function is the action taken when a user invokes this class from the command line.
- Or when run from Eclipse or a program icon on the desktop.
- The program prints **Hello World!** to the screen or terminal.
- How do console programs and graphical programs differ?



Main Components

- Return Value:
 - What the method computes
 - In this case, nothing, so **void**
- Method Name:
 - **main** is reserved for the top-level action
 - Names should describe actions
- Arguments:
 - Specify the input to a method
 - In this case, command line arguments



Main Body

- The method body is a sequence of steps or Java statements
- Statements are separated by semicolons, cannot be omitted!
- In this case, the method has just one statement
- Calls the system object to output a line of text containing **Hello World!**



Practicum

- Use Eclipse to create a workspace, project, and write a program