Eclipse IDE

Lecture 3

What is an IDE?

- Software development involves many different tasks
  - eg Coding, testing, compiling, running, debugging
- Could use a separate tool for each
  - eg Write code with your favorite text editor
- IDE = "Integrated Development Environment"
  - A single tool that supports all these tasks
  - Can address software-specific issues better than separate general-purpose tools (eg text editors)
- Typical features include
  - Syntax highlighting to emphasize structure
  - Code completion ("intellisense")

Eclipse Background

- Origin
  - late 90's: Code base developed by IBM Canada
  - 2001: Released as open source "Eclipse Project"
  - 2004: Formed fully independent industrial consortium, the "Eclipse Foundation"
    - Most developers are employed by member companies
- Many different parts, each is its own product line:
  - Core platform, language tools, plug-ins,...
- Solution: bundled releases (annual)
  - June 2006: "Callisto" (uses 3.2 platform)
  - June 2007: "Europa" (uses 3.3 platform)
  - June 2008: "Ganymede" (uses 3.4 platform)

Eclipse Features

- Multilanguage
  - First love will always be Java
  - Close seconds: C/C++, Web (XML/HTML/CSS)
  - Perl, Python, Ruby, Rails, Mathematica…
- Continuous compilation
  - No compile/build button
  - Syntax/compile-time errors checked as you type
- Extensible through plug-ins
  - About 1000 on www.eclipseplugincentral.com
  - FindBugs: statically identifies possible errors
  - Checkstyle: audits code for “good style” violations
  - djUnit: calculates coverage metrics for test cases

Strengths, weaknesses

- Positives
  - Price
  - Extensibility through plugins
  - Configurability through many preference settings
  - Support for team collaboration
- Negatives
  - Footprint (memory and on disk)

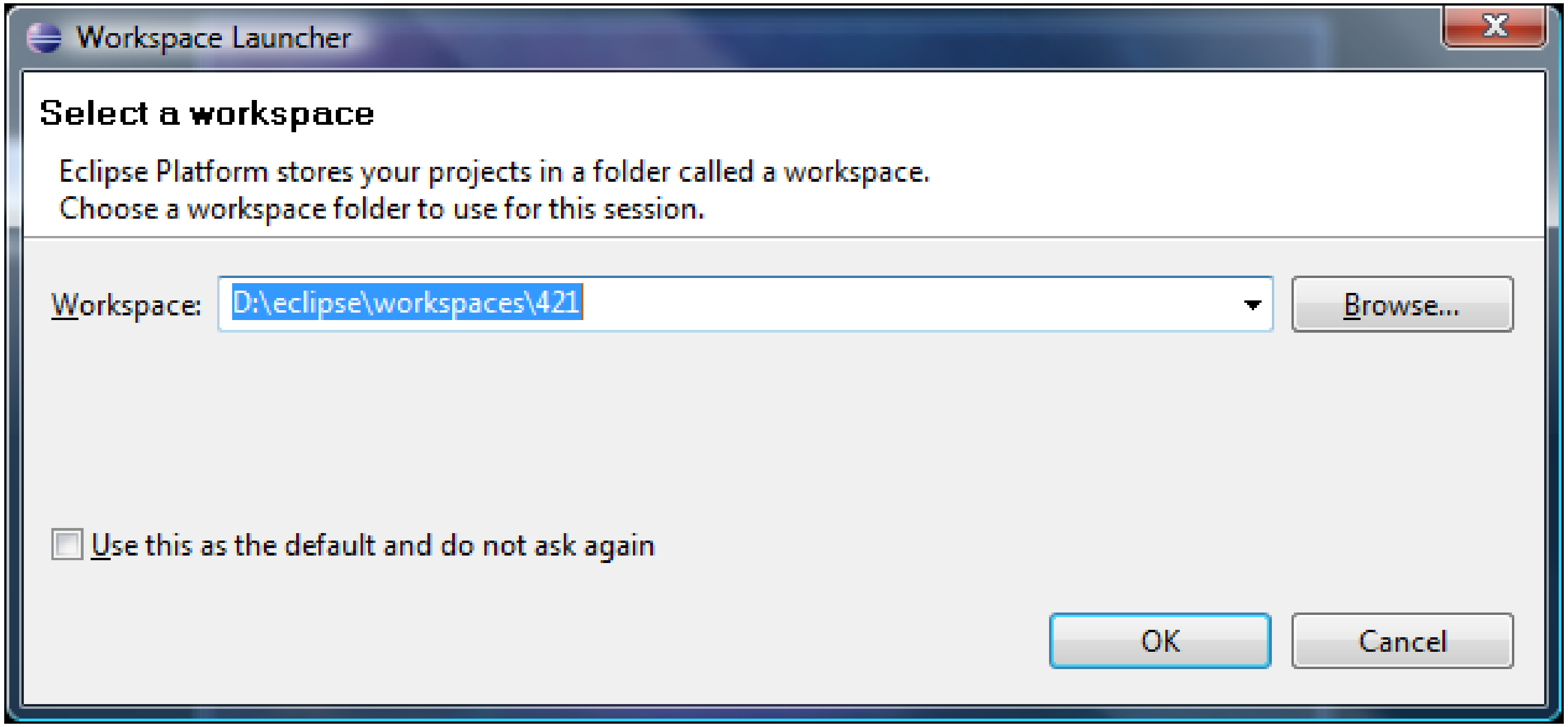
Options for Eclipse Installations

- Windows lab machines (Baker, Caldwell)
  - Eclipse icon on desktop
  - Version 3.3(?); Does not have full JDK!
- Linux login servers
  - See class web site (under “Resources”)
  - Log in to stdlogin using X-Win32 or VNC
  - add /class/cse421/local to your path
  - Run start-eclipse
  - Version 3.4 (and Java JDK 1.6)
- Install at home (Windows, Linux, Mac)
  - See class web site (under “Resources”)
  - Version 3.4 (and Java JDK 1.6)

Workspace

- Eclipse always uses a workspace
  - Prompted at startup for directory
- A workspace contains
  - Projects (ie source files, packages, and libraries)
  - Personal preferences for IDE (eg code formatting)
- Generally need just one workspace
  - Use working sets to reduce clutter
- Multiple workspaces useful for:
  - Multiple Eclipse installations (1 per version)
  - Consultants separating work for multiple clients
- To move preferences between workspaces, use “export”
- For this course, 1 workspace is best choice

Workspace Selection Dialog



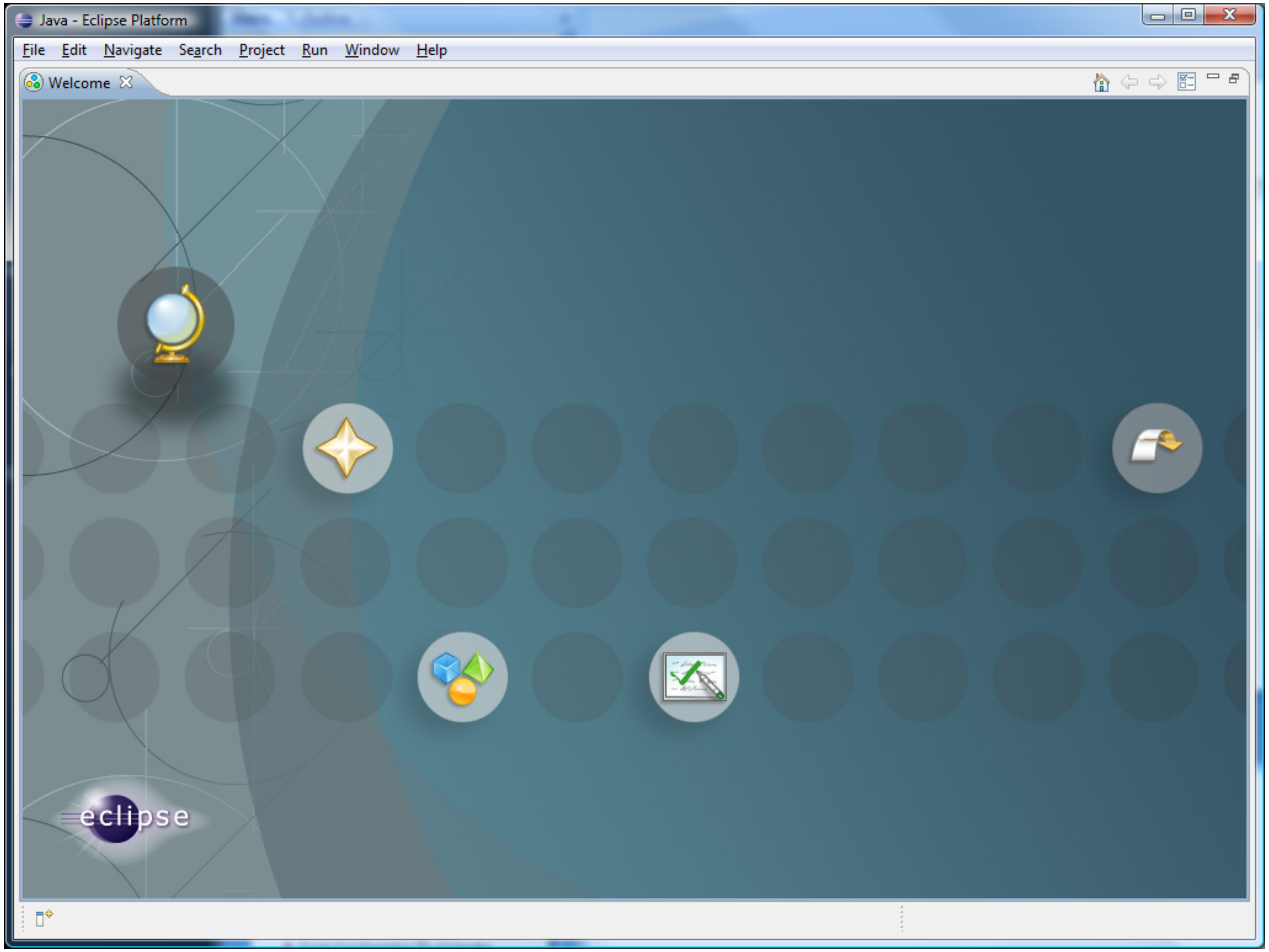
Workspace Launcher

Select a workspace

Eclipse Platform stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: D:\eclipse\workspaces\421

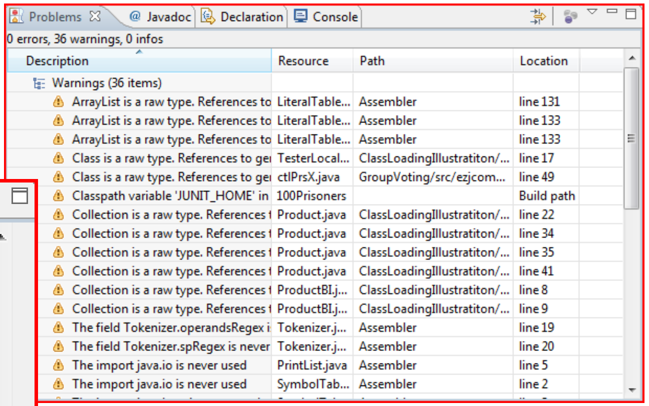
First Launch of Eclipse



Views and Editors

- Building blocks of user interface
- Editor: Associated with input activity
- View: Shows support information
  - Navigate a hierarchy of information
  - Open an editor
  - Display properties for the active editor
- Examples:
  - Java code editor (for writing code)
  - Problems view (compilation errors/warnings)
  - Console view (terminal IO of running program)
  - Class hierarchy view (relating components)
  - Tasks view (todo items)
  - Navigator (file browser)

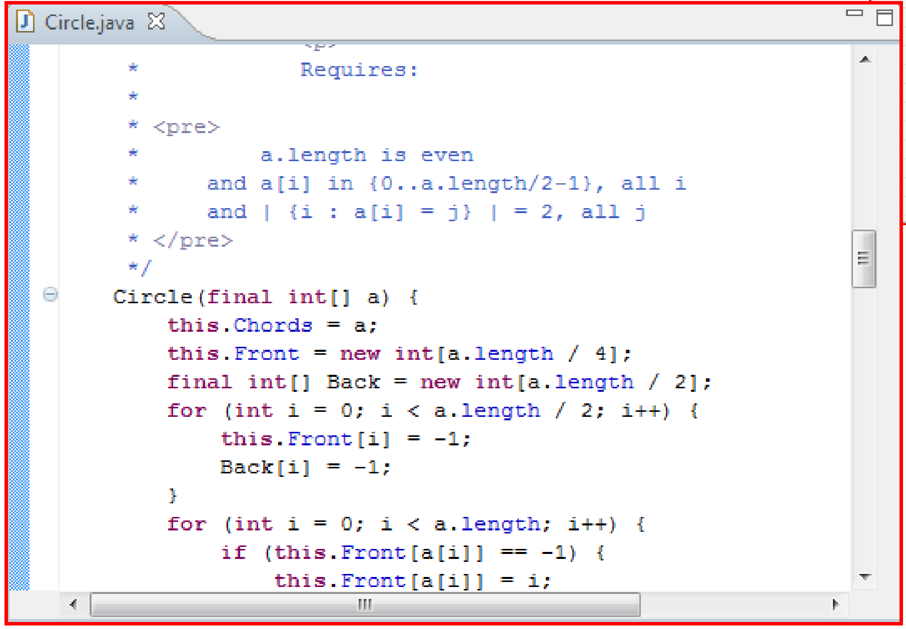
Examples of Editors and Views



0 errors, 36 warnings, 0 infos

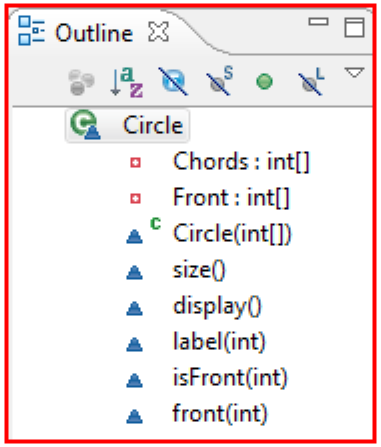
Description	Resource	Path	Location
Warnings (36 items)			
ArrayList is a raw type. References to	LiteralTable...	Assembler	line 131
ArrayList is a raw type. References to	LiteralTable...	Assembler	line 133
ArrayList is a raw type. References to	LiteralTable...	Assembler	line 133
Class is a raw type. References to ge	TesterLocal...	ClassLoadingIllustratiton/...	line 17
Class is a raw type. References to ge	ctlPrsX.java	GroupVoting/src/ezjcom...	line 49
Classpath variable 'JUNIT_HOME' in	100Prisoners		Build path
Collection is a raw type. References t	Product.java	ClassLoadingIllustratiton/...	line 22
Collection is a raw type. References t	Product.java	ClassLoadingIllustratiton/...	line 34
Collection is a raw type. References t	Product.java	ClassLoadingIllustratiton/...	line 35
Collection is a raw type. References t	Product.java	ClassLoadingIllustratiton/...	line 41
Collection is a raw type. References t	ProductBI.j...	ClassLoadingIllustratiton/...	line 8
Collection is a raw type. References t	ProductBI.j...	ClassLoadingIllustratiton/...	line 9
The field Tokenizer.operandsRegex i	Tokenizer.j...	Assembler	line 19
The field Tokenizer.spRegex is never	Tokenizer.j...	Assembler	line 20
The import java.io is never used	PrintList.java	Assembler	line 5
The import java.io is never used	SymbolTab...	Assembler	line 2

Problem View



```
     *            Requires:
     *
     * <pre>
     *          a.length is even
     *      and a[i] in {0..a.length/2-1}, all i
     *      and | {i : a[i] = j} | = 2, all j
     * </pre>
     */
    Circle(final int[] a) {
        this.Chords = a;
        this.Front = new int[a.length / 4];
        final int[] Back = new int[a.length / 2];
        for (int i = 0; i < a.length / 2; i++) {
            this.Front[i] = -1;
            Back[i] = -1;
        }
        for (int i = 0; i < a.length; i++) {
            if (this.Front[a[i]] == -1) {
                this.Front[a[i]] = i;
```

Java Editor



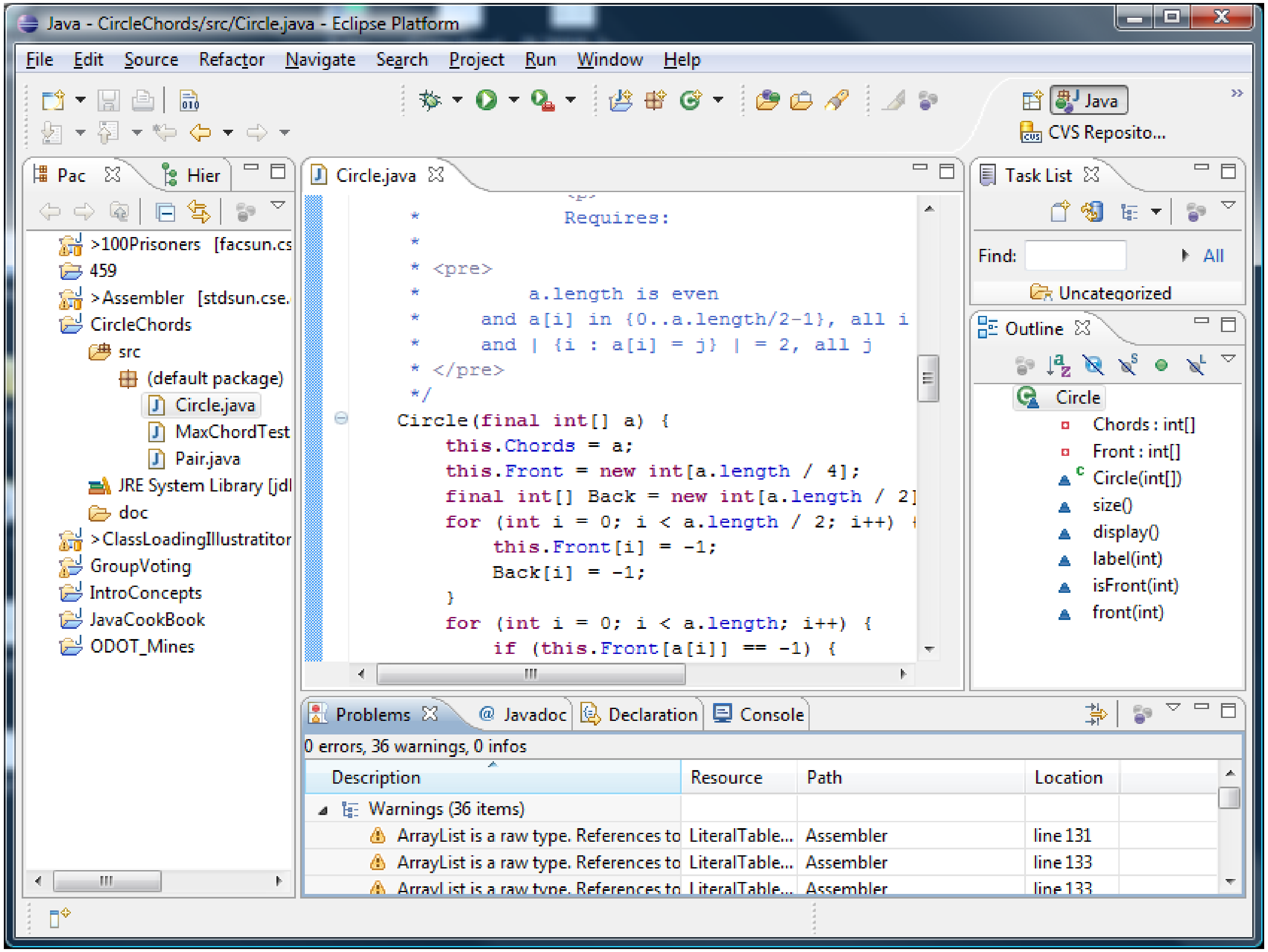
```
Circle
    Chords : int[]
    Front : int[]
    Circle(int[])
    size()
    display()
    label(int)
    isFront(int)
    front(int)
```

Outline View

Perspectives

- A perspective is a particular layout of editors and views
  - Tools are the ones most useful for accomplishing a particular task
- Examples of basic perspectives
  - Java (for writing code)
  - Debug (for debugging a program)
  - Resource (for browsing files)
  - Team Synchronizing (for managing collaborative projects)

Example: Java Perspective



```
 *              Requires:
 *
 * <pre>
 *          a.length is even
 *      and a[i] in {0..a.length/2-1}, all i
 *      and | {i : a[i] = j} | = 2, all j
 * </pre>
 */
Circle(final int[] a) {
    this.Chords = a;
    this.Front = new int[a.length / 4];
    final int[] Back = new int[a.length / 2]
    for (int i = 0; i < a.length / 2; i++) {
        this.Front[i] = -1;
        Back[i] = -1;
    }
    for (int i = 0; i < a.length; i++) {
        if (this.Front[a[i]] == -1) {
```

0 errors, 36 warnings, 0 infos

Description	Resource	Path	Location
Warnings (36 items)			
ArrayList is a raw type. References to	LiteralTable...	Assembler	line 131
ArrayList is a raw type. References to	LiteralTable...	Assembler	line 133
ArrayList is a raw type. References to	LiteralTable...	Assembler	line 133

First Program

- Launch Eclipse
- Open Java perspective
- Create a project
  - File > New > Project, select “Java Project”
  - Name the project (eg HelloWorldProject)
- Create a class within the project
  - File > New > Class
  - Name the class (HelloWorld)
  - Select checkbox to create main()

First Program Continued

- Auto-generates boiler plate code

```
public class HelloWorld {
   /**
    * @param args
    */
   public static void main(String[] args) {
      // TODO Auto-generated method stub

   }

}
```

  - TODO item automatically added to Tasks view
  - Boiler plate comments added too

First Program Continued

- Insert code in main method

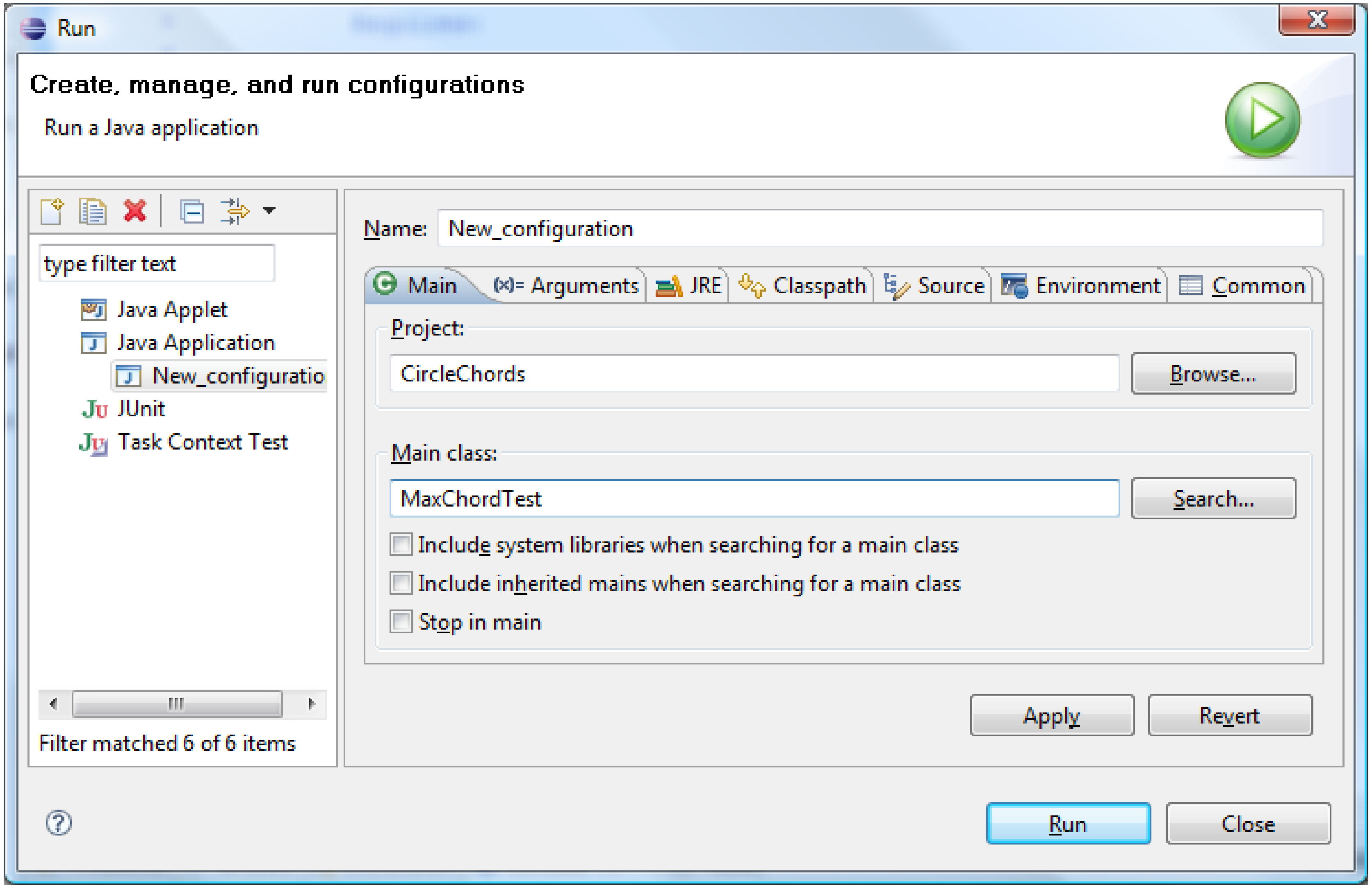
```
System.out.println("Hello World");
```

  - Syntax error appears (temporarily)
  - Command completion after
    - System.
    - System.out.
- Run application
  - Run > Run as > Java application
  - Console view appears with output

Run Configurations

- Controls which project is run and how that project is run
  - Green button runs current file (more or less)
  - Equivalent to "***`$ java classname`***"
- Run > Run Configurations...
  - Select "Java application"
  - Add command-line arguments (see Args tab)
    - Appended to "***`$ java classname`***"
- Advantage: same program can be run in different ways, each stored as its own run configuration

Managing Run Configurations



Run

Create, manage, and run configurations

Run a Java application

type filter text

- Java Applet
- Java Application
  - New_configuratio
- JUnit
- Task Context Test

Filter matched 6 of 6 items

Name: New_configuration

Main | (x)= Arguments | JRE | Classpath | Source | Environment | Common

Project:

CircleChords — Browse...

Main class:

MaxChordTest — Search...

☐ Include system libraries when searching for a main class

☐ Include inherited mains when searching for a main class

☐ Stop in main

Apply | Revert

Run | Close

Personalizing Eclipse

- Window > Preferences
  - General > Appearance > Colors and Fonts
  - General > Editors > Text Editors, Show Line Numbers
  - Java > Code Style / Editor
- Preferences saved in workspace
  - For multiple workspaces, export preferences to a file, then import in other workspace
  - File > Export > General > Preferences
- General advice: avoid tweaking too much

Extending Eclipse

- Easy to install powerful plug-ins
  - recommended: FindBugs, Checkstyle, ECF (already installed for Eclipse on stdlogin)
- Plug-ins can impact performance
  - Consume memory and can slow start up
  - eg Aptana for Web development
- Important advice: Do NOT install plug-ins manually into Eclipse install directory
  - Use Eclipse's installation manager
  - Help > Software Updates
  - Create a "New Site..." and enter the URL for the plugin

Useful Keyboard Shortcuts

- Format: ctrl+shift+F
- Open a class (type): ctrl+shift+T
- Go to current item’s declaration: F3
- Autocomplete suggestions: ctrl+space
- Find references to this item: ctrl+shift+G
- Move between methods: ctrl+shift+up/down
- Go to next error: ctrl+.
- Fix error suggestions: ctrl+1

Supplemental Reading

- Eclipse menu: Help > Welcome
  - Gives original start-up screen with tool overview, tutorials, and code samples
- Eclipse menu: Help > Tips & Tricks...
  - Gives long list of short-cuts and hints
- IBM developerWorks
  - "Getting Started with the Eclipse Platform"
    - http://www.ibm.com/developerworks/opensource/library/os-eclipse-platform/
  - "Introduction to Eclipse for Visual Studio Users"
    - http://www.ibm.com/developerworks/opensource/library/os-eclipse-visualstudio/

Summary

- An IDE supports all code development tasks
- Eclipse basics
  - Installation
  - Workspaces and projects
  - Editors, views and perspectives
- Hello World tutorial with Eclipse
  - Run configurations
- Customization
- Tips & Tricks