

## Choose Your Own Eclipse Plug-In

This project gives students an opportunity to become familiar with open source ideas and to be part of a community. It focuses on Eclipse, an open source community with projects aimed at providing an extensible development platform and application frameworks for building software.

The project team will become a miniature Eclipse developer community that will choose and implement Eclipse extensions. The mission of this experience is to research the capabilities of Eclipse technology, and master what can be done with it. The project members will use their creativity to approach Eclipse technology in new and innovative ways.

During the project, students will learn and have the opportunity to experience important Software Engineering skills such as:

- Working in a team environment
- Collaboration, networking, rapid decision making
- eXtreme Programming
- Design Patterns
- Researching and evaluating technologies
- Connecting with open source developers and industry experts

During the program the students will have an opportunity to learn about Eclipse technologies/subprojects such as:

- Eclipse Modeling Framework (EMF)
- Graphical Editing Framework (GEF)
- Eclipse Communications Framework (ECF)

More Eclipse foundation subprojects can be found at: <http://www.eclipse.org/projects/>

No knowledge of Eclipse is assumed. Java programming skills are required for at least some team members. Design Pattern knowledge will be beneficial. A team of people with various interests and skills will enrich the decision making and collaboration experience.

The project will go through the following stages:

### **Project Initial Stage: What is Eclipse? What is an Eclipse Plug-In?**

- Use the IBM Education Eclipse jump-start resources and other sources to get familiar with Eclipse and plug-in development
- Use online resources to get familiar with eXtreme programming
- Set up a common code sharing framework (e.g. CVS)
- Set up a team newsgroup or forum (e.g. Wiki)

### **Decision Stage: Which plug-ins would be great to implement and have available? Which one will be implemented by the team?**

- Brainstorm ideas: What if we had that cool X plug-in(s)?
- Research what plug-ins already exist
- Document considered choices

- Balance the cool vs. useful, doable vs. desirable factors for the plug-in candidate(s)
- Identify the plug-in of choice

### **Project Idea: Bison/Yacc Eclipse development environment**

- Bison is a free (as in speech and beer) parser generator developed by the GNU open source communities. (<http://www.gnu.org/software/bison/>)
- Antlr, a similar but competing project, already has a very nice Eclipse editor that is not free (as in speech or beer). (<http://www.placidsystems.com/antlrstudio.aspx>)
- Develop a set of Eclipse plug-ins to let you easily edit Bison/Yacc grammars in Eclipse, so possible features are:
  - Syntax highlighting for grammars.
  - Continuous compiling support. (Like what already exists for Java files).
  - Add wizards to generate rules for common actions (e.g. new lines, comments, etc.)
  - Debugger support for grammars that generate Java code. (This should be done in an extensible way so we can add debugger support for other languages later).
  - Visual syntax graphs to help grammar developers see what their grammar is really doing. (I think this could be done using EMF and GMF.)
  - Other that will be useful.
- This project will give lots of exposure to many different areas of software development including:
  - Extensible and interface based architectures.
  - Using many different Eclipse projects and extension points.
  - JNI to interact with some of the Bison libraries.
  - Design patterns so you write good code. ([http://en.wikipedia.org/wiki/Design\\_Patterns](http://en.wikipedia.org/wiki/Design_Patterns))

### **Actual Implementation of an Eclipse Extension**

- Get familiar with the Software Development Cycle and follow it
- Refine functionality requirements
- Create a staged implementation plan for the selected plug-in (if applicable to the selected plug-in)
- Clearly identify team members' roles and dependencies
- Use eXtreme Programming techniques
- Follow relevant Design Patterns
- Team is programming, testing and documenting :)

Throughout the project, team members will perform one or more of the following, based on their interests:

- Contribute to the actual Eclipse code and possibly fix bugs at [bugs@eclipse.org](mailto:bugs@eclipse.org)
- Use the Eclipse.org resources to post technology related questions and ideas (<http://www.eclipse.org/newsgroups/>)
- Use the IBM University Initiative student forums to connect with IBM experts
- Contact other plug-in developers

During the project, all team members will document their experience with Eclipse as an open source community/project and the open source community in general. The team will also document any cool articles, web resources, and books used or considered for the project. As part of the final project deliverables the team will write an Eclipse.org article about their experience with Eclipse and Eclipse plug-ins.

As part of the final deliverables, the project team is also encouraged to create a presentation on Eclipse technology and share their experience with the Computer Science department.

### **IBM contacts**

Project Lead: Gergana Markova  
Main Technical Mentor: Mark Weaver  
Technical Mentor: Chris Aniszczyk  
Project Mentor: Christy Lauridsen

### **References**

Eclipse.org

<http://www-128.ibm.com/developerworks/opensource/top-projects/eclipse-starthere.html?ca=dgr-lnxw09LearnEclipse>

<http://www.extremeprogramming.org>

### **Consideration**

:) Opportunity is missed by most people because it is dressed in overalls and looks like work.  
Thomas A. Edison