Programming Language:

Espresso!

a small cup of Java

PLT CS 4115
Prof. Stephen Edwards
September 23, 2003

Joya Zuber (jaz49), Philip Coakley (pwc35),
Aaron Bauman (adb54), Erin Adelman (eca41)

1. Introduction and Overview:

The *Espresso!* programming language facilitates the creation of Java applets, providing an introduction to the basics of Java syntax and semantics. As computing saturates many aspects of our lives today, programming aptitude is becoming a significant part of what it means to be computer literate. With *Espresso!* inexperienced programmers and children can learn to quickly create powerful applets. This small cup of java offers a positive and practical start to a student's computer education.

2. Background:

Learning to build graphical java applications without prior programming experience is like running a marathon without training. *Espresso!* was created in order to achieve results without treading through the overhead associated with a robust Java education. Realizing that it is possible (as well as easy and fun) to use the computer as a development tool is an important step in the training of the novice programmer.

Apple's Hypercard serves as a simple interface to create dynamic multimedia presentations. Hypercard uses the easy to understand metaphor of cards and stacks to aid the user in quick comprehension of the program's functionality. The intuitive nature of Hypercard has made it a successful educational tool. In addition to the card and stack metaphor, a drag and drop interface allows novice users to realize graphical applications. However, this design choice does not forward the developer's programming prowess.

A forerunner in the cross-platform, programming, and object-oriented paradigm, Java is a necessary component of modern computer science education. In order to learn Java, one needs to become comfortable with the use of a vast API and complicated structure. This can be intimidating to the beginning programmer.

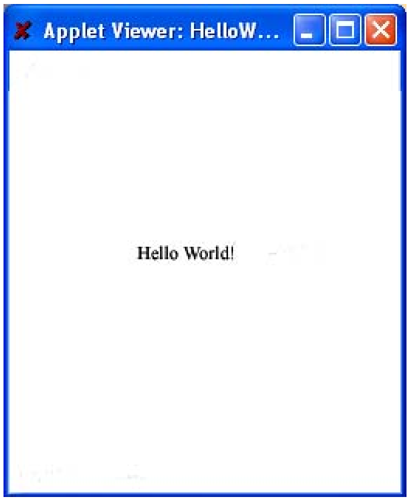
Espresso! is a fusion of the pedagogical nature of Hypercard and the power of Java.

3. Simple:

Simplicity is a defining feature of *Espresso!* For the beginning programmer, *Espresso!* affords the functionality of Java without the tedious and confusing overhead involved in creating a Java applet from scratch. The easy-to-use primitives with pre-programmed defaults provide quick production of graphical interfaces. For example, note the simplicity of a basic HelloWorld program in *Espresso!* :

```
card HelloWorld {
      Textbox myText(“Hello World!”);
}
```

If, as above, no other parameters are specified, *Espresso!* will center this black text on a white card, and add the card to the a default stack. A typical Java program would take approximately 10 lines to create the resulting *Espresso!* applet below:



Hello World!

Espresso! simplifies algorithmic tasks such as for loops. The code below demonstrates an initialization loop for a group of 10 buttons:

```
card SomeButtons {
      ButtonGroup myButtons[10];
      do(i,10) {
            myButtons[i].text = “button” + i;
      }
}
```

5. Object-Oriented:

The card and stack analogy lends itself to an inherently object-oriented design. By grouping graphical and textual elements into card objects, *Espresso!* familiarizes students with the basics of object-oriented design in an intuitive and logical manner.

6. Portable:

Espresso! code compiles to create Java applets, exploiting the renowned portability of Java. *Espresso!* is as transportable as any standard Java application, and viewable in all Java-enabled web browsers.

8. Educational:

A goal of *Espresso!* is to provide a spring board to the world of Java programming. Its instructional intentions place education at the heart of the language. *Espresso!* makes a great first language for any child or inexperienced programmer!