**Tutorial 1**

In this section we will create a simple program in the Next programming language that outputs “Hello world”. Keep in mind that in this step by step each line individually explained is part of a single program. The Next programming language can be thought of as a language made up of declarations. A programmer can declare a number or a string. We will begin by declaring an integer and setting its value to 0:

int num = 0;

We will then declare a string:

string announce = “Not in the world”;

In Next we can also declare items, characters and locations. Here we will declare a character and a location. What follows is a character declaration. The first parenthesis contains the character’s attributes while the second contains the items the character is carrying. We will leave the second parenthesis empty because our character is not carrying anything:

character you {(string slogan = “Hello world!”),()}

What follows is a location declaration. The first parenthesis contains the location’s attributes, the second contains the items the location contains and the third the characters the location contains. We will leave the first two empty and only include a character in the location declaration. This code looks as follows:

location here {(),(),(you)}

There is a special type of declaration in next in which we declare an event to start. Once a location has been declared, a start declaration must be made for that location. In order to use a location in conjunction with a start statement the location must be declared first. In order to use a character in a location that character must be declared first. In order to use an item in a character or a location the item must be declared first. This is why in the Next language it is a good programming practice to create all of the item, character and location declarations first and in that order followed by the start declarations. The start declarations must contain the programming logic for each location because aside from other declarations Next does not allow statements to be placed outside of a start declaration. The start location we will use for this program looks as follows:

start here end (num == 1) {

if (exists here.you) then

output you.slogan;

else

output announce;

num = 1;

}

This start declaration starts the **here** location and specifies that it will end when integer **num** is equal to 1. The expression

if (exist here.you) then

means if character **you** exists in location **here** then execute the code that follows. In this case if true then the **slogan** attribute from the **you** character is output with the line

output you.slogan;

Else the string **announce** is output. To finish execution of the **here** start statement integer **num** is set equal to 1 to meet the end condition. If this had not been done the start statement would iterate forever over its body of code. Because **you** exists in location **here** the program will output “Hello World!” Here is the complete program:

int num = 0;

string announce = "not in the world:";

character you {(string slogan = "Hello world!"),()}

location here {(),(),(you)}

start here end (num == 1) {

if (exists here.you) then

output you.slogan;

else

output announce;

num = 1;

}

**Compiling and Running a Next Program**

Once you have created a next program save it with a .next file extension. Go to where the Next file is located. Use the Next file and your .next extension file as input and output the results into a file called Next.java. The command to do this would look like this

**$ ./Next < Tutorial1.next > Next.java**

In this example the Next source code file is Tutorial1.next which is located in the same folder as the Next file. This outputs a java file which can then be compiled and executed as any other java source code file. For example to compile the file you could use the command

**$ javac Next.java**

Then to run the file you could use

**$ java Next**