
For each

COMP 102

Victoria University of Wellington

For loop statement

Common scenario

- a list of values
- actions to perform for each value in the list

// print each number in a list of numbers:

```
for ( int i=0; i < listOfNumbers.size(); i++ ) {  
    UI.println(listOfNumbers.get(i));  
}
```

// print each string in a list of numbers that starts with "A":

```
for (int i=0; i < listOfStrings.size(); i++ ) {  
    if (listOfStrings.get(i).startsWith("A") ) {  
        UI.println(listOfStrings.get(i));  
    }  
}
```

For Each statement

Common scenario

- a list of values
- actions to perform for each value in the list

// print each number in a list of numbers:

```
for ( double num : listOfNumbers ) {  
    UI.println(num);  
}
```

listOfNumbers:

num:

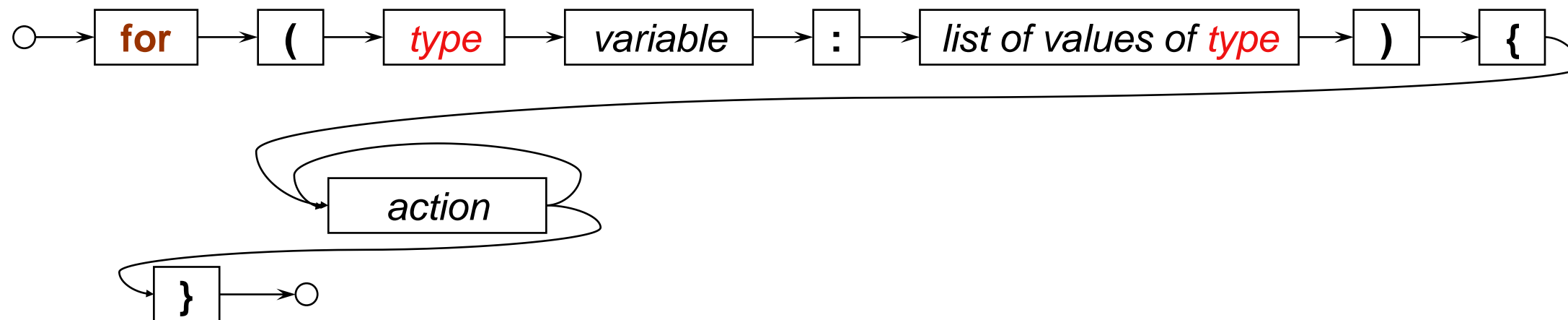
// print each string in a list of numbers that starts with "A":

```
for ( String str : listOfStrings ) {  
    if ( str.startsWith("A") ) {  
        UI.println(str);  
    }  
}
```

listOfStrings:

str:

For statement ("for each" version)



```

for ( double num : listOfNumbers ) {
    UI.println(num);
}

```

- Meaning:

Repeatedly (for each value in the list)

- put the next value of the list into the variable
- do the actions.

Lists of values

- What type is a **list** of values?
- How do we get a list of values?

List of doubles

Have to use `Double`, not `double`
`Double` is the "wrapped-up" version of `double`,
 for putting into a list

```
ArrayList <Double> numberList = UI.askNumbers("Enter numbers");
```

```
for (double num : numberList) {
    UI.println(num);
}
```

Asks for a list of numbers, ending with 'done'

```
UI.setColor(Color.red);
UI.setLineWidth(5);
```

```
for (double radius : numberList) {
    if (radius > 20 && radius < 200) {
        UI.drawOval( 300 - radius, 250 - radius, radius * 2.0, radius * 2.0);
    }
}
```

Lists of values

- What type is a **list** of values?
- How do we get a list of values?

List of String values

```
ArrayList <String> nameList = UI.askStrings("Enter names");
```

```
for (String name : nameList) {
    UI.println("Hello " + name);
}
```

Asks for a list of strings, ending with empty line

```
UI.println("==== Long names =====");
```

```
for (String name : nameList) {
    if (name.length() > 6 ) { UI.println(name); }
}
```

```
UI.println("==== Short names =====");
```

```
for (String name : nameList) {
    if (name.length() <= 6 ) { UI.print(name + ", "); }
}
```

print without a new line

```
UI.println();
```

print just a new line

Doing more with the loops: using Variables

- Add up all the numbers in a list:

numberList: 150.0, 32.2, 6.9, 49.5, 83.4, -21.0, 1.0

```
ArrayList <Double> numberList = UI.askNumbers("Enter numbers");
```

Declare and initialise variable

```
double total = 0.0;
```

total: 300.0

```
for (double num : numberList) {
```

num: 150.0

```
    total = total + num;
```

```
}
```

```
UI.println("Total of numbers = " + total );
```

Add each number into the total:

- Uses current value in total
- Adds the next number to it
- Puts result back into total

Doing more with the loops: using Variables

- Count the number of long names in a list.

```
ArrayList <String> nameList = UI.askStrings("Enter names");
```

```
int count = 0;
```

Declare and initialise variable

```
for (String name : nameList) {
```

```
    if (name.length() > 6 ) {
```

```
        count = count + 1;
```

Add 1 to the count

```
    }
```

```
}
```

```
UI.printf("There were %d long names out of %d names \n", count, nameList.size() );
```

Number of values in a list

Lists are values too: passing lists around

```

public void analyseNames() {
    ArrayList <String> nameList = UI.askStrings("Enter names");
    UI.println("Total characters: " + this.totalChars (nameList) );
    UI.println("Starts with A: " + this.wordStartingWith(nameList, "A" ) );
}

public int totalChars(ArrayList <String> strings ){
    int count = 0;
    for (String str : strings) {
        count = count + str.length();
    }
    return count;
}

public String wordStartingWith(ArrayList <String> strings, String pattern ){
    for (String str : strings) {
        if ( str.startsWith(pattern) ) { return str; } // returns first word starting with the pattern
    }
    return "<none>";
}

```