



Bilkent University

Department of Computer Engineering

CS 315 Programming Languages

Design a Drawing Figure Language

Part I Tutorial

Project Group Member Name:

Emir Acımiş, 21201233

Izel Gürbüz, 21301018

Mustafa Culban, 21301187

Supervisor: Ramazan Gökberk Cinbiş

Language Name: Drawer++

Nov 28, 2016

1-Entry Point

Drawer++ supports main as the entry point function that starts by writing “begin”. Programmer must write the code into the section between “begin” and “end” to be able to run.

Example Code:

```
begin
....
....

end
```

2-Variable Names

Variable names are to reach the variable after creating from somewhere else of the code. Variable names start with lowercase letter and except first one , every word starts with uppercase letter in the name. The name supports just alphanumeric characters.

3-Integer

Integers are being used directly without any special notation in Drawer++. Also variables can be assigned as integer to be able to use later. It is done by just writing “integer” before variable name and then programmer can give a value to this variable by putting equal character “=” between variable name and the value or leave the variable just as unknown-valued integer. Furthermore, an integer variable with value can be used to define the other integer variable’s value and value of an integer variable can be change somewhere else in the code.

Example Code:

```
integer int1
```

```
integer int2
```

```
int2 = 987
```

```
int1 = int2
```

```
int2 =654
```

4-Float

Floats are being used directly without any special notation in Drawer++. Also variables can be assigned as float to be able to use later. It is done by just writing “float” before variable name and then programmer can give a value to this variable by putting equal character “=” between variable name and the value or leave the variable just as unknown-valued float. Furthermore, a float variable with value can be used to define the other float variable’s value and value of a float variable can be change somewhere else in the code.

Example Code:

```
float f1
```

```
float f2
```

```
f2 = 987.654
```

```
f1=f2
```

```
f2=32.1
```

5-String

Strings are being used by writing the string between “” characters in Drawer++. Also variables can be assigned as string to be able to use later. It is done by just writing “string” before variable name and then programmer can give a value to this variable by putting equal character “=” between variable name and the value or leave the variable just as empty string. Furthermore, a string variable with value can be used to define the other string variable’s value and value of a string variable can be change somewhere else in the code.

Example Code:

```
string str1
string str2

str2 = "a text"
str1=str2
str2="another text"
```

6-Basic Arithmetic Operations

Drawer++ supports basic arithmetic operations as summation, subtraction , multiplication and division. To calculate the result of mathematical expression , expression must be written directly on a line. These operations are used with only float and integer values/variables. Also, Drawer++ allows operations with both integer and float values/variables. When programmer uses only integer , result becomes an integer value (some cases for division and multiplication result is shown as rounded properly).If programmer uses just floats or both integers and floats , result becomes a float value.

Example Code

```
integer int1 = 12
float    f1   = 1.2

int1 + 3.4
int1 - 56
f1   * 7.8
f1   / 9
```

7-Logical Operations

Drawer++ supports logical indicators as “==”(equal indicator), “!=“(not being equal indicator), “>”(greater than indicator), “<” (smaller than indicator), “>=”(greater than or equal indicator) , “<=” (smaller than or equal indicator). Logical operations must be written directly in the parentheses “()” and gives an output in the boolean type. Logical indicators “!=“ and “==” are used with float, integer or string types of variables although other indicators are used with float or integer values. In a logical expression, programmer can use float and integer values together but just string values with other string values.

Example Code :

```
integer int1 = 1
float   f1   = 2.3
float   f2   = 4.5
```

```
(int1 <= 3 )
(f1   == f2)
(f2   >  int1)
(“hello” != “Hi”)
```

8-Instantiating Shapes

All shapes are instantiated same way that is writing “Shape” at the beginning of the line then variable name then “=” notation followed by the specified shape construction with or without parameters. If the shape will have the default scale , programmer doesn’t write any parameter. Otherwise, programmer must write arbitrary parameters meeting with feature requirements of the specified shape.

Example Code:

```
Shape s1, s2
Location l1 = Location (10,20)
Color c1    = Color (RGB{0,0,225,0.8})
integer w1  = 10
```

```
integer sw1 =15
integer h1   = 5
Set a1       = {l1.xValue,l1.yValue,w1,h1}
```

```
s1 = Rectangle()
s2 = Oval(c1,a1,sw1)
```

9-Drawing Shapes

Drawing shapes is possible when an instance of the shape type is created, by functions. Also , by different functions , shapes can be drawn with default sizes or arbitrary values of sizes on an arbitrary location.

```
Shape s1 = Oval()
draw.s1
```

10-Functions

To call a function programmer must write function name then a dot and at the end the instance of the shape will be drawn.

Example Code:

```
Shape s1 = Line()
draw.r1
```

11-Comments

Comments are used with the // notation at the beginning of the line . This notation makes whole line the comment line for compiler.

Example Code:

```
// First comment line
```

```
...
```

```
...
```

```
//Second comment line
```

```
..
```

12-Set

Set is represented by variables/values separated with comma between curly parentheses.

In Drawer++ there are Set of color and attributes.

Example Code:

```
Location l1 = Location (10,20)
```

```
integer w1 = 10
```

```
integer sw1 =15
```

```
integer h1 = 5
```

```
Set a1 = {l1.xValue,l1.yValue,w1,h1,sw1}
```

13-Loop

Drawer++ has a loop called “For” . For loop provides repetition of the operations including curly parenthesis until statement in the parenthesis just after “for” is not valid anymore.

The statement consists of three part separated by semicolons. First part presents starting condition , second part includes the condition checked to decide repeating , last part represents changes before next repetition.

Example Code :

```
....
```

```
for(integer a)= 0 ; a< 10; a++)
```

```
{
```

```
.....
```

```
.....
```

```
}
```

14-if - and if - else

Drawer++ supports conditionals as “if” , “and if” , “else”. When programmer must check a accuracy of a statement to make some operations, as boolean type of result , this statement should be written in the parenthesis with “if” at the beginning and operations done when condition identified in the statement, should be written between curly parenthesis. If there are just two possible case as the statement is valid or not , operation (if there is) done when the identified condition is not valid , should be written after writing ”else”. If there are more than two case , except first one , cases should be identified in the parenthesis after writing “and if”.

Example Code:

```
.....  
if(4>5)  
{  
.....  
}  
and if(3==7)  
{  
.....  
}  
else{  
.....  
}
```