

# Assignment One

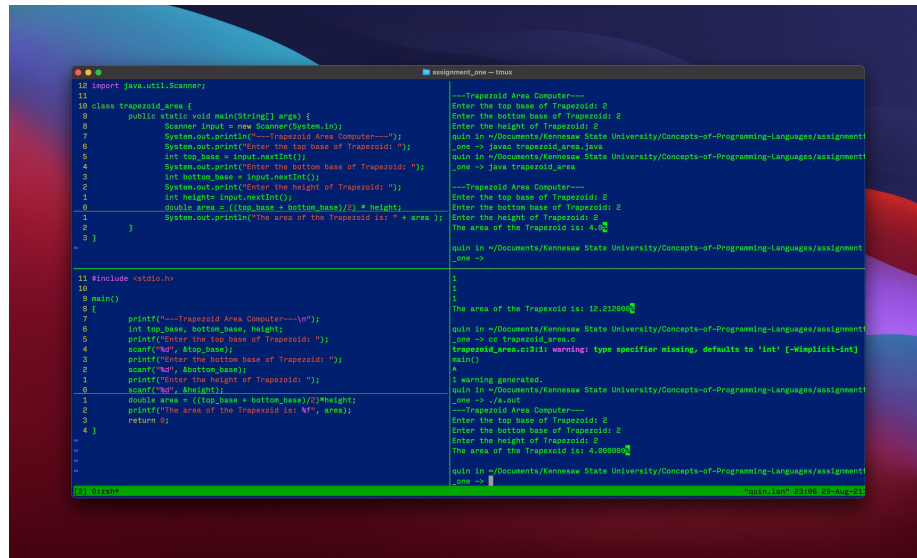
Quin'darius Lyles-Woods

August 30, 2021

# Problem One

Compiling and running programs on the command line.

- Open a command line window and compile the Java program with the 'javac' compiler.
- Verify that you have a compiled program using the 'dir' command (on Windows).
- Run the compiled program using the 'java' virtual machine.
- Compile and link the C++ program using the GNU C++ compiler.
- Verify that you have an executable file of the program.
- Execute the executable program.
- Show all your work.



```
12 import java.util.Scanner;
13
14 class trapezoid_area {
15     public static void main(String[] args) {
16         Scanner input = new Scanner(System.in);
17         System.out.println("Trapezoid Area Computer---");
18         System.out.print("Enter the top base of Trapezoid: ");
19         int top_base = input.nextInt();
20         System.out.print("Enter the bottom base of Trapezoid: ");
21         int bottom_base = input.nextInt();
22         System.out.print("Enter the height of Trapezoid: ");
23         int height = input.nextInt();
24         double area = ((top_base + bottom_base)/2) * height;
25         System.out.println("The area of the Trapezoid is: " + area);
26     }
27 }

11 #include <stdio.h>
12
13 int main()
14 {
15     printf("Trapezoid Area Computer---\n");
16     int top_base, bottom_base, height;
17     printf("Enter the top base of Trapezoid: ");
18     scanf("%d", &top_base);
19     printf("Enter the bottom base of Trapezoid: ");
20     scanf("%d", &bottom_base);
21     printf("Enter the height of Trapezoid: ");
22     scanf("%d", &height);
23     double area = ((top_base + bottom_base)/2)*height;
24     printf("The area of the Trapezoid is: %f", area);
25     return 0;
26 }
```

Terminal output for Java:

```
---Trapezoid Area Computer---
Enter the top base of Trapezoid: 2
Enter the bottom base of Trapezoid: 2
Enter the height of Trapezoid: 2
The area of the Trapezoid is: 4.0
quit in ~/Documents/Kennesaw State University/Concepts-of-Programming-Languages/assignment1_one -> java trapezoid_area.java
quit in ~/Documents/Kennesaw State University/Concepts-of-Programming-Languages/assignment1_one -> java trapezoid_area
```

Terminal output for C++:

```
1
1
1
The area of the Trapezoid is: 12.012800
quit in ~/Documents/Kennesaw State University/Concepts-of-Programming-Languages/assignment1_one -> cc trapezoid_area.c
trapezoid_area.c:31:1: warning: type specifier missing, defaults to 'int' [-Wimplicit-int]
main()
^
1 warning generated.
quit in ~/Documents/Kennesaw State University/Concepts-of-Programming-Languages/assignment1_one -> ./a.out
---Trapezoid Area Computer---
Enter the top base of Trapezoid: 2
Enter the bottom base of Trapezoid: 2
Enter the height of Trapezoid: 2
The area of the Trapezoid is: 4.000000
quit in ~/Documents/Kennesaw State University/Concepts-of-Programming-Languages/assignment1_one ->
```

## Problem Two

Rewrite the BNF of Example 3.4 to give + precedence over \* and force + to be right associative.

$$\begin{aligned} \langle assignment \rangle &\mapsto \langle id \rangle = \langle expression \rangle \\ \langle id \rangle &\mapsto A \\ \langle id \rangle &\mapsto B \\ \langle id \rangle &\mapsto C \\ \langle expression \rangle &\mapsto \langle expression \rangle * \langle term \rangle \\ \langle expression \rangle &\mapsto \langle term \rangle \\ \langle term \rangle &\mapsto \langle factor \rangle + \langle term \rangle \\ \langle term \rangle &\mapsto \langle factor \rangle \\ \langle factor \rangle &\mapsto \langle expression \rangle \\ \langle factor \rangle &\mapsto \langle id \rangle \end{aligned}$$

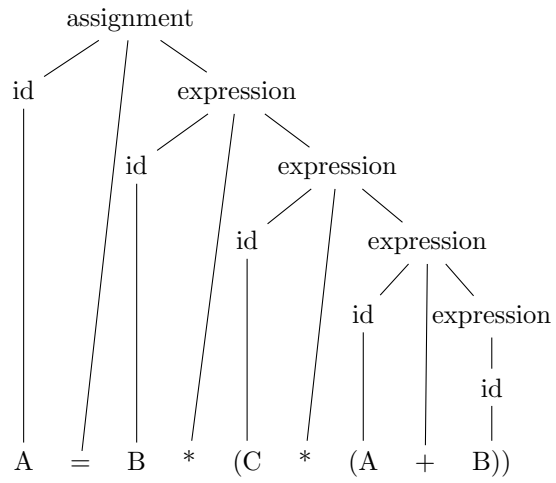
## Problem Three

Using the grammar in Example 3.4, show a parse tree and a leftmost derivation for each of the following statement:  $A = B * (C * (A + B))$

### Derivation

$$\begin{aligned} \langle assignment \rangle &\mapsto \langle id \rangle = \langle expression \rangle \\ &\mapsto A = \langle expression \rangle \\ &\mapsto A = \langle id \rangle * \langle expression \rangle \\ &\mapsto A = B * \langle expression \rangle \\ &\mapsto A = B * (\langle expression \rangle) \\ &\mapsto A = B * (\langle id \rangle * \langle expression \rangle) \\ &\mapsto A = B * (C * \langle expression \rangle) \\ &\mapsto A = B * (C * (\langle expression \rangle)) \\ &\mapsto A = B * (C * (\langle id \rangle + \langle expression \rangle)) \\ &\mapsto A = B * (C * (A + \langle expression \rangle)) \\ &\mapsto A = B * (C * (A + \langle id \rangle)) \\ &\mapsto A = B * (C * (A + B)) \end{aligned}$$

## Parse Tree



made proudly with