

CSE 1720

Building Interactive Systems

1



Chapter 1

Introduction to Programming

2

Outline

1.1 Anatomy of a Program

- 1.1.1 A Quick Tour
- 1.1.2 Language Elements
- 1.1.3 Program Execution

1.2 The Declaration Statement

- 1.2.1 Variable Names
- 1.2.2 The Integer Types
- 1.2.3 Declaration and Memory
- 1.2.4 Other Data Types
- 1.2.5 Primitive and Non-Primitive Types

1.3 The Assignment Statement

- 1.3.1 The int Arithmetic Operators
- 1.3.2 Other Arithmetic Operators
- 1.3.3 Mixed Types and Casting

3

1.1.1 A Quick Tour

Let us take a look at a Java program. It does not matter now how the program was written; just become familiar with the terminology for describing program structures.

Note, in particular, the following four terms:

Imports, Class, Method, Style

4

```
import java.lang.System;
public class Area
{
    public static void main(String[] args)
    {
        int width;
        width = 8;
        int height = 3;
        int area = width * height;
        System.out.println(area);
    }
}
```

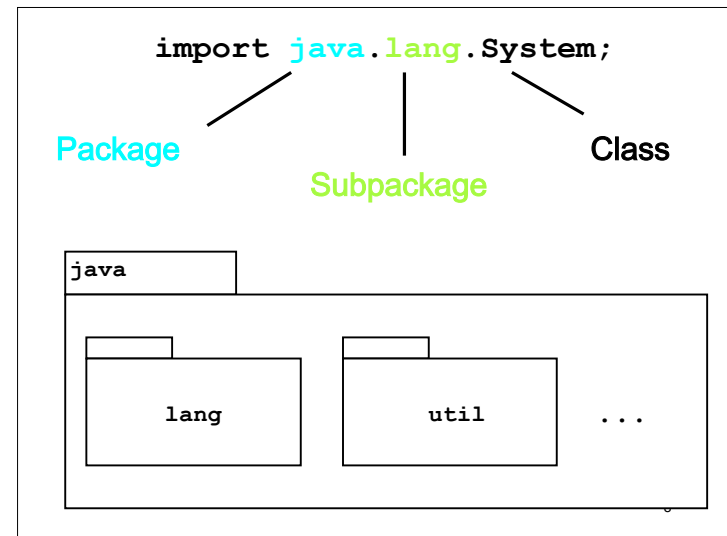
5

```
import java.lang.System;
public class Area
{
    public static void main(String[] args)
    {
        int width;
        width = 8;
        int height = 3;
        int area = width * height;
        System.out.println(area);
    }
}
```

6

```
import java.lang.System;
public class Area
{
    public static void main(String[] args)
    {
        int width;
        width = 8;
        int height = 3;
        int area = width * height;
        System.out.println(area);
    }
}
```

7



- Note the difference:

```
import java.lang.System;
versus
import java.lang.*;
```

- And as a matter of style:

The package naming convention calls for lowercase letters.

9

```
import java.lang.System;
public class Area
{
    public static void main(String[] args)
    {
        int width;
        width = 8;
        int height = 3;
        int area = width * height;
        System.out.println(area);
    }
}
```

Class Header

Class Body, a Block

10

```
import java.lang.*;
public class Area
{
    public static void main(String[] args)
    {
        int width;
        width = 8;
        int height = 3;
        int area = width * height;
        System.out.println(area);
    }
}
```

Method Header

Method Body, a Block

11

Style

Class naming convention

Use title case unless an acronym, e.g. Math, URL, StringTokenizer.

Method naming convention

Use lowercase letters but for multi-word names, capitalize the first letter of each subsequent word, e.g. main, equals, toString, isLeapYear

Block layout

Braces must align vertically and the all statements must be left justified and indented by one tab position.

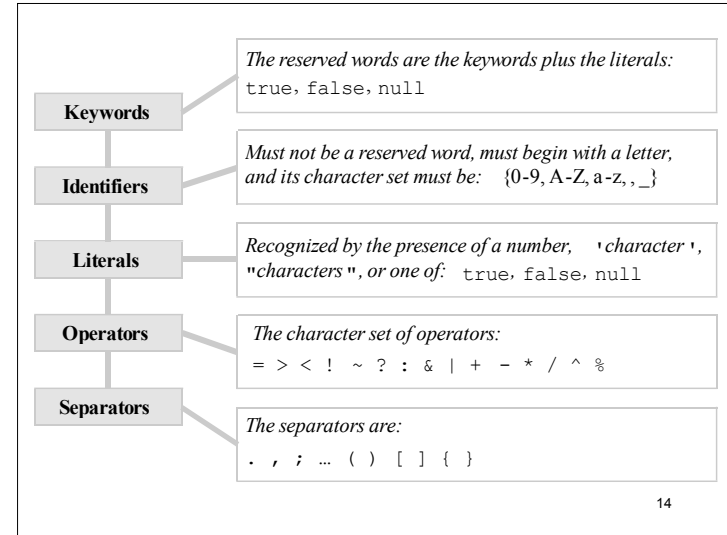
12

1.1.2 Language Elements

Still without worrying about semantics, let us identify the elements of a program:

- Keywords
- Identifiers
- Literals
- Operators
- Separators

13



14

Keywords

abstract	assert				
case	catch	char	class	const	continue
else	enum	extends			
goto					
long					
package	private	protected	public		
short	static	strictfp	super	switch	synchronized
void	volatile				

15

Example

Identify the language elements in the following program...

Keywords, Identifiers, Literals, Operators, Separators

16

```
import java.lang.System;

public class Area
{
    public static void main(String[] args)
    {
        int width;
        width = 8;
        int height = 3;
        int area = width * height;
        System.out.println(area);
    }
}
```

Keywords, Identifiers, Literals, Operators, Separators

17

```
import java.lang.System;

public class Area
{
    public static void main(String[] args)
    {
        int width;
        width = 8;
        int height = 3;
        int area = width * height;
        System.out.println(area);
    }
}
```

Keywords, Identifiers, Literals, Operators, Separators

18

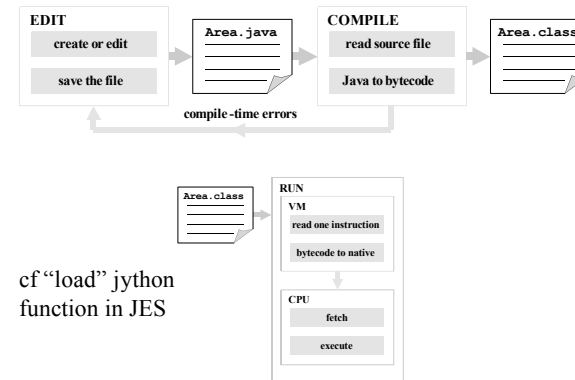
```
import java.lang.System;

public class Area
{
    public static void main(String[] args)
    {
        int width;
        width = 8;
        int height = 3;
        int area = width * height;
        System.out.println(area);
    }
}
```

Keywords, Identifiers, Literals, Operators, Separators

19


1.1.3 Program Execution



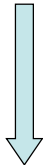
20

1.2 The Declaration Statement


type name ;



The name of a primitive or non-primitive type, e.g. int, double...



An identifier to be associated with a memory block



A separator

The **scope** of the variable = the enclosing block of the declaration. The variable is not known outside its scope.

21

1.2.1 Variable Names

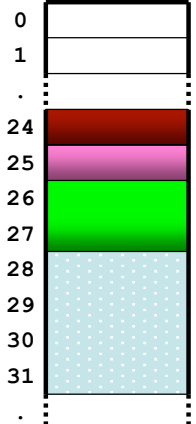
Rules and guidelines for the name:

- Must be an **identifier**
- Must not be in the **scope** of another variable with the same name
- A good name is **indicative** of the content that will be stored in the variable
- As a matter of style, use **lowercase** letters, but for multi-word names, capitalize the first letter of each subsequent word

22

1.2.3 Declaration and Memory

- Logical versus Physical
- Address versus Content
- Bytes, KB, MB, GB, TB, ...
- **The Memory Block**
 - 1-byte block at address 24
 - 1-byte block at address 25
 - 2-byte block at address 26
 - 4-byte block at address 28

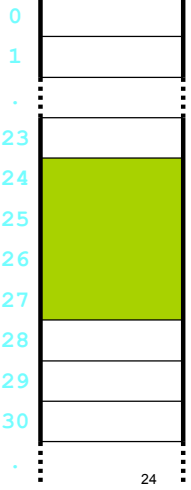


23

What happens when we write:

int width;

width →



1. A block big enough to hold an **int** is allocated, e.g. a 4B block at 24
2. Its address is associated with the variable name, e.g. 24 with width

Note that no initialization is involved; only an association of a name with an address.

24