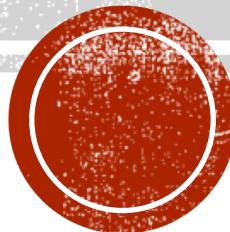




Middlesex
University
London

Lecture 2

Input and Output



REVISION TASK

- Write a Java program which would print your Uni time table to the console;
- The program must have correct class declaration, main method and it has to use just one of the print methods, including escape sequences for editing the output.
- Ideally it should also have a statement which would add all of your teaching hours

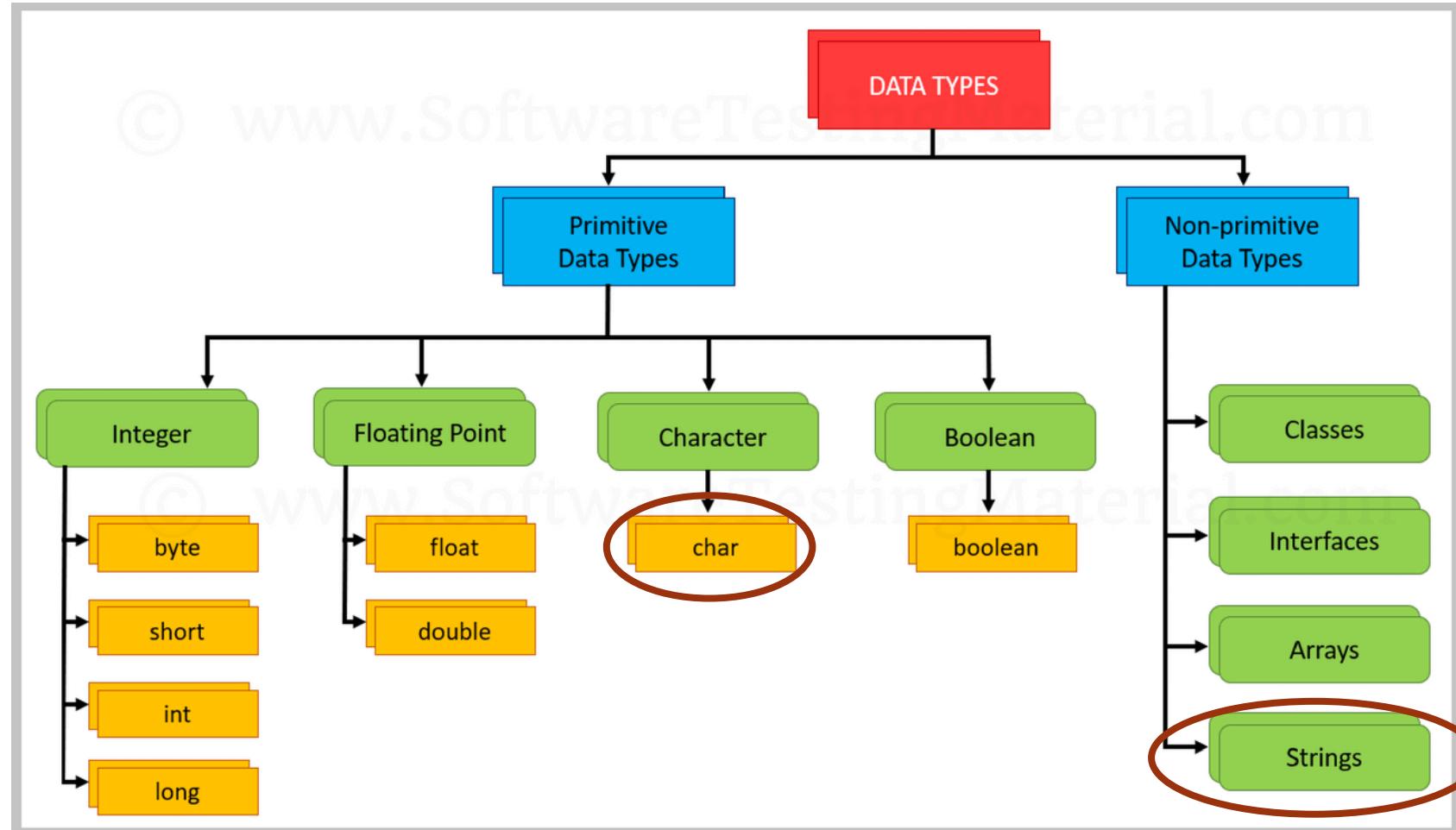


THIS WEEK

- Variable
- Char primitive variable
- String variable
- Concatenation
- Scanner
 - Input – Output



DATA TYPES



Today we would be looking at `char` and `String`.
The rest of the primitive Types we will cover next week.

VARIABLE

- A variable is a name given to a memory location.
- It is the basic unit of storage in a program.
- Each variable has specific data type which determines size of memory, name and value
- The value stored in a variable can be changed during program execution.
- A variable is only a name given to a memory location, all the operations done in the variable effects that memory location.
- In Java, variables must be declared before it can be used.



VARIABLE DECLARATION

Type: defines what type of data would be stored in variable.

Identifier/Variable Name: is used to differentiate between different variables in a program.
Variable name identifiers use the camel-case naming convention and starts from small letter.
(choose meaningful names)

Assignment operator:
used to assign value to variable

Semicolon:
is used as terminator of the statement

Value that is being assigned to a variable

CHAR

- **char** is a primitive data type for declaring characters
- Variable declared as char can hold single character such as 'j' or '\$'
- We define char in Java program using single quote (')
- Java uses the UNICODE standard to store character data, 16 bits (2^{16} or 65536)
- ASCII is an early standard that encodes characters as numbers, 8 bits (2^8 or 256)
 - so the 1st 256 characters from the Unicode character set are compatible with the ASCII character set
 - Internally, a char is stored as number
- Example:

```
//defining variable  
char at;  
//assigning value  
at = '@'; }  
  
char at = '@';  
  
//printing value of the variable  
System.out.println(at);
```



STRING

- The String data type is for declaring a sequence of characters
- We can define String in Java using double quotes (""), such as "Java"
- String can have none or more characters, including: empty string, a blank, number, symbol and escape character
- **String** is not-primitive data type (Strings are objects not variables)
 - An object consist of data that belongs to the object and methods (actions) that the object can do (Java.lang package)
- Example:

```
//defining variable  
String fruit;  
//assigning value  
fruit = "Apple "; }  
  
//printing value of the variable  
System.out.println(fruit);  
String fruit = "Apple";
```



TASK

- Create a new project and declare all the variable that we have cover until this point



CONCATENATION

- (as an adjective): from late Latin concatenat- ‘linked together’, from the verb concatenare, from con- ‘together’ + catenare, from catena ‘chain’.
 - en.oxforddictionaries.com

- The **+** operator has two meanings :
 1. for addition, 2. concatenate two string together

- Strings also provide
concat method to concatenate two strings.

```
String word1 ="snow";
String word2 ="ball";
String word = word1 + word2;
System.out.println(word);
```

```
String word1 ="snow";
String word2 ="ball";
String word = word1.concat(word2);
System.out.println(word);
```

At least one of the operand must be a string in order for concatenation to take place

If number is concatenated with a string the number is converted into string and concatenated



PRACTICE

- What would be a result of the following concatenations?
 - `System.out.println("4"+"4");`
 - `System.out.println("Hello"+"World");`
 - `System.out.println("Middlesex "+"University");`
 - `System.out.println('a'+'b');`



METHODS IN CHARACTER CLASS

Method	Description
Character.isDigit(ch);	Return true if the specific character is digit
Character.isLetter(ch)	Return true if the specific character is letter
Character.isLetterOrDigit(ch)	Return true if the specific character is letter or digit
Character.isLowerCase(ch)	Return true if the specific character is lower case
Character.toUpperCase(ch)	Return the upper case of the specific character
Character.toLowerCase(ch)	Return the lower case of the specific character



METHODS IN CHARACTER CLASS- EXAMPLE:

```
char size = 's';

System.out.println("It is: "+ Character.isDigit(size) +" that the character \'"+ size + '\'' is digit: " );
System.out.println("It is: "+ Character.isLetter(size) +" that the character \'"+size+"\' is letter");
System.out.println("It is: "+ Character.isLetterOrDigit(size) +" that the character \'"+size+"\' is letter or digit");
System.out.println("It is: "+ Character.isLowerCase(size) +" that the character \'"+size+"\' is lower case");
System.out.println("The character \'"+size+"\' is upper case: "+ Character.toUpperCase(size));
System.out.println("The character \'"+size+"\' is lower case: "+ Character.toLowerCase(size));
```

run-single:

```
It is: false that the character 's' is digit:
It is: true that the character 's' is letter
It is: true that the character 's' is letter or digit
It is: true that the character 's' is lower case
The character 's' is upper case: S
The character 's' is lower case: s
```



METHODS IN STRING CLASS

Method	Description
length()	Return the number of characters in this string
charAt(index)	Return the character at the specific index from this string
concat()	Return a new string that concatenates this string with string s1
toUpperCase()	Return a new string with all letters in uppercase
toLowerCase()	Return a new string with all letters in lowercase
trim()	Return a new string with whitespace characters trimmed on both side



METHODS IN STRING CLASS - EXAMPLE

```
String title = " Introduction to Java Programming ";

System.out.println("The number of characters in the title "+title+" is: "+title.length());//return the length of the title string
System.out.println("The first letter in the title: "+title + " is "+ title.charAt(0));// because we are counting strings from 0
System.out.println("The author of the Java book is: "+title.concat(" is: Y. Liang ")); //the concat function concatenates two strings together
System.out.println("The title is: "+title.toUpperCase());
System.out.println("The title is: "+title.toLowerCase());
System.out.println(title.trim());//eliminates leading and trailing spaces
```

Output:

```
The number of characters in the title Introduction to Java Programming is: 34
The first letter in the title: Introduction to Java Programming is
The author of the Java book is: Introduction to Java Programming is: Y. Liang
The title is: INTRODUCTION TO JAVA PROGRAMMING
The title is: introduction to java programming
Introduction to Java Programming
BUILD SUCCESSFUL (total time: 0 seconds)
```



GETTING CHAR FROM STRING

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	e	l	c	o	m	e	t	o		J	a	v	a	

```
welcome.length(); //15
```

```
String welcome = "Welcome to Java";
```

```
System.out.println("The String Length "+ welcome.length()); //15
```

```
System.out.println("The first letter is: "+welcome.charAt(0)); //W
```

```
System.out.println("The last letter is: "+ welcome.charAt(welcome.length()));//?
```



PRACTICE

- In NetBeans declare a char with one character and a string with short sentence and practice all the basic methods we have just covered.



SCANNER

- `java.util.Scanner;`
 - Enables a program to read data for use in a program.
 - Data can come from many sources, such as input through the keyboard or a file on a disk.
 - To use Scanner we need to create an object of it:

```
type name = new type();
```

The `new` keyword creates an object.

`System.in` refers to the standard input device
- the keyboard

```
Scanner input = new Scanner(System.in);
```



SOME SCANNER METHODS

```
//importing util package
import java.util.Scanner;

public class test {

    public static void main(String[] args) {

        // creating a Scanner object
        Scanner input = new Scanner(System.in);

        //Used to receive next token.
        //next() can read the input only till the space. It can't read two words separated by a space.
        input.next();

        //Used to receive a char variable
        input.next().charAt(0);

        //Used to receive next String token
        input.nextLine();

    }
}
```



SCANNER EXAMPLE

```
// creating a Scanner object
Scanner input = new Scanner(System.in);
System.out.println("What country are you coming from? ");//prompt
String country = input.nextLine(); //country as a String

System.out.print("Enyter you name:");//prompt
String name = input.next(); // name is a String

System.out.print("Enter the 1st letter or the symbol of currency used in "+ country); //prompt
char currency = input.next().charAt(0); //currency as one char

System.out.println("in "+country + ", "+name+ " is using "+ currency);
```



TASK

- Create new project where you would generate a student email. Ask user for the first part of his/her email and concatenate it to the listed below variables in order to create one email.

```
char at = '@';  
String endOfStudentEmail = "live.mdx.ac.uk";
```

Output:

The Student email is: aw674@live.mdx.ac.uk



KEY THINGS YOU SHOULD UNDERSTAND AFTER WEEK 2

- Variable
- Char primitive variable
- String variable
- Concatenation of a string with the '+' symbol
- Reading input from the console with a Scanner
- Naming conventions



MASTERING YOUR SKILLS

- Read from page 147-155
- Make sure you finish all your SOBs.

