Char and String

Sungchul Lee

Learning Object

- Char
  - ASCII Encoding
  - Character Processing
- String
  - Create a String object
  - String Concatenation

Char (Characters)

- **char** data type is used to represent a single character
- Characters are stored in a computer memory using some form of **encoding**
  - Java uses Unicode, which includes ASCII, for representing **char** constants
  - *ASCII* (*American Standard Code for Information Interchange)*, is one of the document coding schemes widely used today

ASCII Encoding

	0	1	2	3	4	5	6	7	8	9
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht
10	lf	vt	ff	cr	so	si	dle	dc1	dc2	dc3
20	cd4	nak	syn	etb	can	em	sub	esc	fs	gs
30	rs	us	sp	!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	}	|	~	del		

For example, character 'O' is 79 (row value 70 + col value 9 = 79).

Unicode Encoding

- The Unicode Worldwide Character Standard (Unicode) supports the interchange, processing, and display of the written texts of diverse languages.
- Java uses the Unicode standard for representing char constants.

```
char ch1 = 'X';

System.out.println(ch1);                    X
System.out.println( (int) ch1);             88
```

Character Processing

```
char ch1, ch2 = 'X';
```

Declaration and initialization

```
System.out.print("ASCII code of
character X is " + (int) 'X' );

System.out.print("Character with
ASCII code 88 is " + (char)88 );
```

Type conversion between int and char.

```
'A' < 'c'
```

This comparison returns true because ASCII value of 'A' is 65 while that of 'c' is 99.

String

- What is the “string”?
  - “I am a Java Programmer”, “12345”
  - String a = “I love JAVA”;
  - String b = new String(“I love JAVA”);
- A string is a sequence of characters that is treated as a single value
- Instances of the String class are used to represent strings in Java
  - String is class/object, not a primitive data type

Create a String object

- Syntax:

  String <variable name>;

  <variable name>=**new** String(“<value of a string>”);

  Example:

  ```
  String strVar;
  strVar = new String(“CS 172 Course”);
  ```

  OR

  String <variable name>;

  <variable name> = “<value of a string>”;

  ```
  String strVar = “CS 172 Course”;
  ```

String Concatenation

- Method: (add two strings)

  Create a new string from two strings by ***concatenating*** the two strings

- Example:

```
String strVar1 = “CS 172”;
String strVar2 = “Course”;
String sumStr;

sumStr = strVar1+strVar2;
```

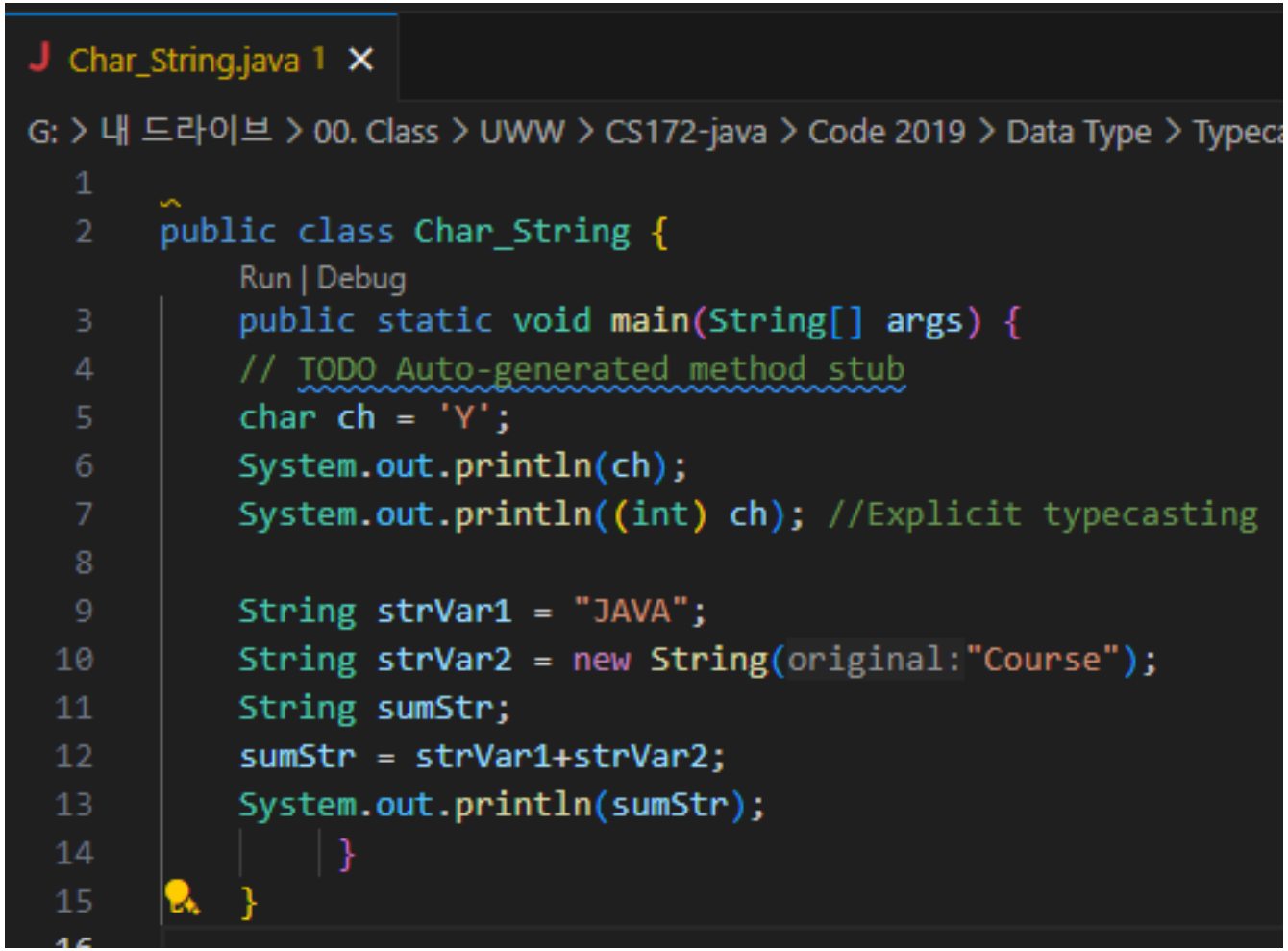
Practice

1. Make a new project (Reference: Create Project and Class File)
   - ❑Project name: Char_String
2. Create a new Class File
   - ❑Class name: Char_String
3. Coding:

```
public class Char_String {
        public static void main(String[] args) {
        // TODO Auto-generated method stub
        char ch = 'Y';
        System.out.println(ch);
        System.out.println((int) ch); //Explicit typecasting

        String strVar1 = "CS 172";
        String strVar2 = new String("Course");
        String sumStr;
        sumStr = strVar1+strVar2;
        System.out.println(sumStr);
    }
}
```

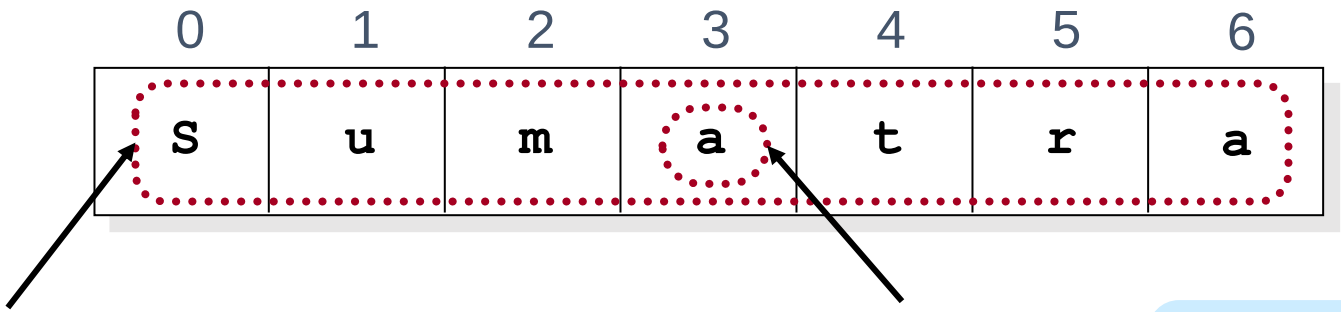
Practice – Code



Accessing Individual Elements

- Individual characters in a String accessed with the charAt method.
  - The index starts at 0

```
String name = "Sumatra";
```



This variable refers to the whole string.

The method returns the character at position # 3.

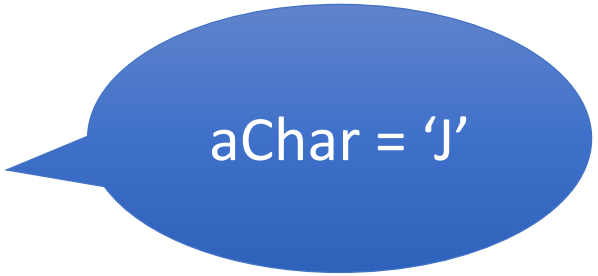
Character at position k of a string

Example:

String sample=“CS JAVA Course”;

char aChar;

aChar = sample.charAt(3)



Compute Length of a string

- Method: length()
  - Returns the length of a string
    - Space also counts
    - String str =“123 456”; // length 7

- Example:

```
String strVar;
strVar = new String(“CS JAVA Course”);
int len = strVar.length();
```

String Comparison

- Methods:
  - equals
  - equalsIgnoreCase
  - contains
- Return **true** if two strings are the **same**, **false if not**
  - String string1 =“CS JAVA”;
  - String string2 = “JAVA”;
  - String string3 = “cs JAVA”;
  - Boolean isEqual1, isEqual2,isEqual3;
  - isEqual1 = string1.equals(string2);
  - isEqual2 = string1.equalsIgnoreCase(string3);
  - isEqual3 = string1.contains(string2)

Practice – String

```
-

false
true
true
9
false
```

Char_String.java 1

G: > 내 드라이브 > 00. Class > UWW > CS172-java > Code 2019 > Data Type > Typec

```java
public class Char_String {

    Run | Debug
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        String course = "CS - Java";
        String othercourse = "CS - JAVA";

        System.out.println(course.charAt(index:3));
        System.out.println(course.charAt(index:4));

        boolean check = course.equals(othercourse);
        System.out.println(check);
        check = course.equalsIgnoreCase(othercourse);
        System.out.println(check);
        check = course.contains(s:"Java");
        System.out.println(check);

        int len = course.length();
        System.out.println(len);
        System.out.println (true &&false);

    }

}
```

Summary

- Char
  - ASCII Encoding
  - Character Processing
- String
  - Create a String object
  - String Concatenation