



ALZEEKA Tutorial

Programming 1 – CSCE 101



053 359 7191



<https://alzeeka.github.io/alzeeka/>

FREE

ملخص 3 (class String) lecturer

*الكلمات الي تحتها خط ركزون عليها

❖ Precedence Rules

High precedence



First: The unary operators: +, -, ++, --, and !

Second: The binary arithmetic operators: *, /, And %

Third: The binary arithmetic operators: + and -

Low precedence

*قواعد الأسبقية

- تبدأ أولا من العمليات الأحادية مثلا سالب العدد او موجب العدد والزيادة والنقصان
- و ثم علامات الضرب والقسمة وباقي القسمة
- واخيرا علامات الجمع والطرح

*ولا ننسى انه يمكن استخدام الأقواس من اجل التحكم بالأسبقية ودائما اذا صادفت علامات لهم نفس الأسبقية ابدأ حل من اليسار الى اليمين



053 359 7191



***أمثلة:**

افترض ان $x=2$ وال $y=4$ وال $z=5$ اوجد الناتج

- $x + y * z \rightarrow 2 + 4 * 5 \rightarrow 2 + 20 = 22$
- $x + (y * z) \rightarrow 2 + (4 * 5) \rightarrow 2 + 20 = 22$
- $3x \rightarrow 3 * 2 = 6$
- $3x + y \rightarrow 3 * 2 + 4 \rightarrow 6 + 4 = 10$
- $\frac{3x+y}{z+2} \rightarrow (3 * 2 + 4) / (5+2) \rightarrow 10 / 7 = 1$



❖ GPA:

من خلال الجدول هذا يمكنك معرفة الدرجات مع المعدل لهذه الدرجات , لانه ربما تحتاجه في الدروس القادمة

Grades in English	Points	Marks	Letter Grades
Exceptional	4.00	95 – 100	A+
Excellent	3.75	90 – Less than 95	A
Superior	3.50	85 – Less than 90	B+
Very Good	3.00	80 – Less than 85	B
Above Average	2.50	75 – Less than 80	C+
Good	2.00	70 – Less than 75	C
High-Pass	1.50	65 – Less than 70	D+
Pass	1.00	60 – Less than 65	D
Fail	0.00	Less than 60	F
In Progress	-	-	IP
Incomplete	-	-	IC
Denial	0.00	-	DN

Quality Points		GPA		Code	%	Cr Hrs	Course
7	9	3.50	4.50	B+	85	2	IAS 301
6	9	2.00	3.00	C	70	3	CHEM 324
11.25	14.25	3.75	4.75	A	92	3	MATH 235
12	16	3.00	4.00	B	80	4	PHYS 312
36.25		48.25		12			Total

يمكن حساب المعدل من خلال هذا القانون : مجموع النقاط على عدد الساعات

$$GPA = \frac{\text{Total Quality Points (36.25)}}{\text{Total Credits (12)}} = 3.02$$



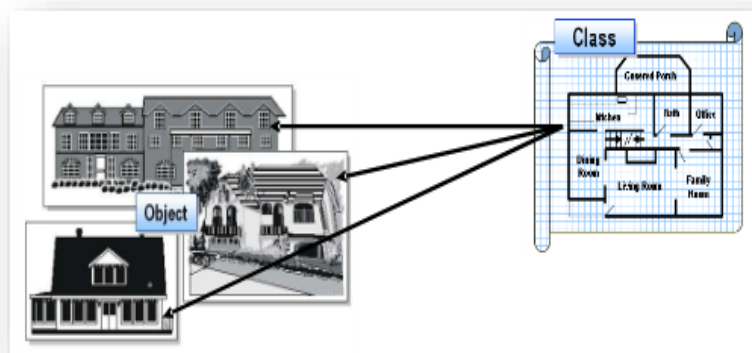
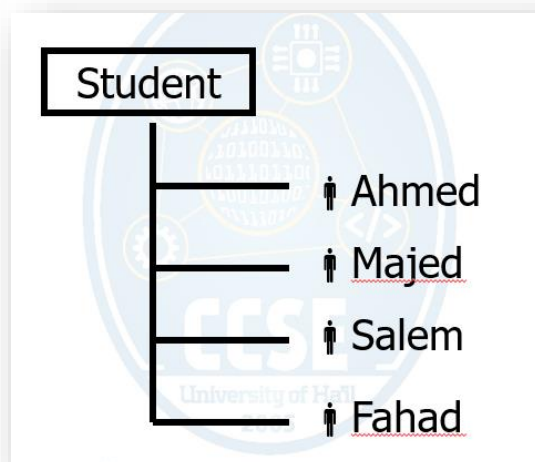
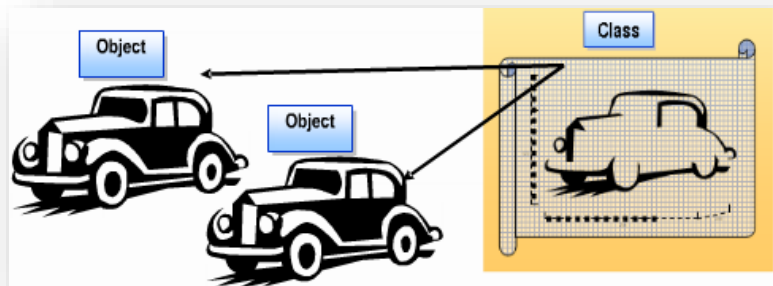
053 359 7191



❖ String type is a Class

ما هو الفرق ما بين **Object** & **class** ؟

بإختصار ال **class** يعتبر قالب او مخطط وال **object** يعتبر نسخة من هذا ال **class** ,
شوف المثال التالي معنا مثلا مخطط لسيارة ومن خلال هذا المخطط صنعنا عدة
سيارات و الصورة الثانية نفس المثال.



نفس المثال بالجافا نفترض انه معنا كلاس **String** ولديه عدة **object** مثلا **name1,2,3**

```
String name1 = "ahmed";  
String name2 = "mohammed";  
String name3 = "khlaed";
```



❖ The Class String

- There is no primitive type for strings in Java
- The class String is a predefined class in Java that is used to store and process strings
- String يعتبر نوع no primitive يعني نوع غير بدائي ويستخدم لتخزين ومعالجة البيانات النصية
- Objects of type String are made up of strings of characters that are written within double quotes
"Hello, how are you?"
- A variable of type String can be given the value of a String object
String sentence = "Java is Fun!";
- ويجب دائما ان اجعل البيانات النصية بين (double quotes) علامتي التنصيص " "

❖ Classes, Objects, and Methods

- A **class** is the name for a type whose values are objects
- **Objects** are entities that store data and take actions
- The action/s that an object can take are called **method**
- **الكلاس** هو اسم لنوع ما تكون قيمه أشياء (objects).
- **objects** هي كيانات تخزن البيانات وتنفيذ إجراءات.
- الإجراءات التي يستطيع objects تنفيذها تسمى **دوال**.
- **Example:**
String name;
name.length();
 - Class is **String**. Object is **name**. Method is **length**.



❖ String Methods examples

1. String Concatenation
2. String Length
3. String Starting Position
4. Other Useful String Operators

1. Concatenation of Strings الدمج او الربط

Concatenation: Using the + operator on two strings in order to connect them to form one longer string

- اذا معي سلسلتين نصيتين وعشان اسوي ربط او **Concatenation** راح استخدم علامة + ومن خلال هذه العلامة سيكون الناتج سلسلة نصية واحدة, شوف هذا المثال :

```
String greeting = "Hello ";
String javaClass = "class";
String word = greeting + javaClass;
// word = "Hello class"
```

- اذا معي **greeting = "Hello"** وال **javaClass = "class"** وسويت **word = greeting + javaClass** راح تكون قيمة ال word هي **"Hello class"**

- Any number of strings can be concatenated together
- When a string is combined with almost **any other type** of item, the result is a **string**
- برضو يمكن عمل ربط ال string مع الأرقام وناتج أي ربط لأي نوع مع ال string سيكون الناتج **string**

"The answer is " + 42 evaluates to "The answer is 42"



```
String greeting = "Hello ";  
String course = "ICS102";  
System.out.println(greeting + course);
```

Hello ICS102

```
int x = 6, y = 10;  
String s = "JAVA";  
System.out.println(s + x);  
System.out.println(x + y + s);  
System.out.println(s + (x + y));  
System.out.println(s + x + y);
```

JAVA6
16JAVA
JAVA16
JAVA610

• *ملاحظات :

- اذا صادفت جمع عددين قبل ال string راج تجمع طبيعي مثل

`System.out.println(x + y + s);`

- اذا صادفت جمع عددين بداخل قوس بعد ال string راج تجمع طبيعي مثل

`System.out.println(s + (x + y));`

- اذا صادفت جمع عددين بدون قوس بعد ال string هنا راج تسوي ربط وتعاملهم كأنهم string مثل

`System.out.println(s + x + y);`



2. String Length (طول السلسلة النصية)

- We determine the number of characters in a String with the **length method**.
- من خلال الدالة **length method** يمكن حساب عدد الحروف في السلسلة النصية

```
String name = "WELCOME";  
String str2 = "";  
String str3;
```

```
name.length();
```

7

```
str2.length();
```

0

```
str3.length();
```

Error!

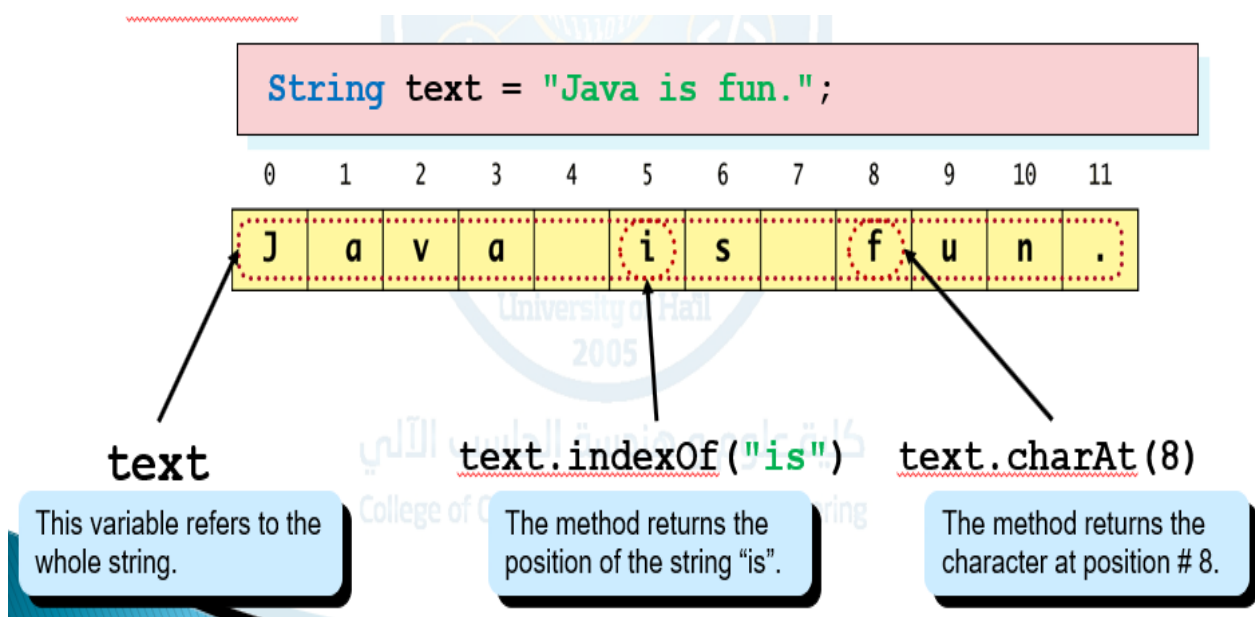
Error because no data is stored for str3, so it is a **null**.

- في اخر سطر راج يظهر error لأن قيمة str3 فارغة او **null**



3. String Starting Position

- Individual characters in a String can be accessed with the **charAt method**.
- Position of a character or String can be found with **indexOf method**.
- **charAt method** تستخدم لكي اوصل لحرف معين
- **indexOf method** تستخدم لكي اوصل لموقع حرف او سلسلة نصية معينة.
- ولكي اتعامل مع الدالتين لازم استخدم واعرف ال **index** حيث يبدأ من الصفر , يعني لما يكون **index = 0** كأنني ماسك اول حرف من السلسلة النصية



- **charAt method** تحتاج argument بين القوسين يكون رقم ال **index** لاحظ انه الحرف الموجود في **index 8** هو ال **f**
- **indexOf method** تحتاج argument بين القوسين يكون اما حرف او string وهي راج تجيب لي رقم ال **index** للحرف او لبداية السلسلة النصية.



4. Other String Methods

Method	Type	Returns Value
<code>length()</code>	<code>int</code>	Returns the length
<code>equals(Str2)</code>	<code>boolean</code>	Returns true if <code>str1= str2</code> . Otherwise, returns false
<code>equalsIgnoreCase(Str2)</code>	<code>boolean</code>	uppercase and lowercase of a letter to be the same
<code>toLowerCase()</code>	<code>String</code>	Returns a string with all letter lowercase.
<code>toUpperCase()</code>	<code>String</code>	Returns a string with all letter uppercase.
<code>trim()</code>	<code>String</code>	Returns a string with white space removed.
<code>charAt(Position)</code>	<code>char</code>	one character . Positions are counted 0, 1, 2, etc.
<code>substring(Start)</code>	<code>String</code>	substring from Start to the end of string
<code>substring(Start, End)</code>	<code>String</code>	Returns the substring from Start to the End
<code>indexOf(A_String)</code>	<code>int</code>	Returns the position of the first occurrence of the string. Returns -1 if <code>A_String</code> is not found.

❖ Some Methods in the Class String

```
int length()
```

Returns the length of the calling object (which is a string) as a value of type `int`.

EXAMPLE

After program executes `String greeting = "Hello!";`
`greeting.length()` returns 6.

- هذي الدالة تحسب لي طول او عدد الحروف (مع الرموز والمسافات) للسلسلة النصية وهي من نوع `int`



053 359 7191



`boolean equals(Other_String)`

Returns true if the calling object string and the *Other_String* are equal. Otherwise, returns false.

EXAMPLE

After program executes `String greeting = "Hello";`
`greeting.equals("Hello")` returns `true`
`greeting.equals("Good-Bye")` returns `false`
`greeting.equals("hello")` returns `false`

Note that case matters. "Hello" and "hello" are not equal because one starts with an uppercase letter and the other starts with a lowercase letter.

- هذي الدالة تقارن بين سلسلتين نصيتين وهي من نوع **Boolean** , (حساسة لحالة الأحرف) يعني اذا معي "Hello" ومعني "hello" راج يكون **false** لانهم يختلفون بأول حرف.

`boolean equalsIgnoreCase(Other_String)`

Returns true if the calling object string and the *Other_String* are equal, considering uppercase and lowercase versions of a letter to be the same. Otherwise, returns false.

EXAMPLE

After program executes `String name = "mary!";`
`greeting.equalsIgnoreCase("Mary!")` returns `true`

- هذي الدالة تقارن بين سلسلتين نصيتين وهي من نوع **Boolean** , (ليست حساسة لحالة الأحرف) يعني اذا معي "Hello" ومعني "hello" راج يكون **true** .



String toLowerCase()

Returns a string with the same characters as the calling object string, but with all letter characters converted to lowercase.

EXAMPLE

After program executes `String greeting = "Hi Mary!";`
`greeting.toLowerCase()` returns `"hi mary!"`.

- هذي الدالة تحول كل السلسلة النصية الى **حروف صغيرة small** وهي من نوع **String**

String toUpperCase()

Returns a string with the same characters as the calling object string, but with all letter characters converted to uppercase.

EXAMPLE

After program executes `String greeting = "Hi Mary!";`
`greeting.toUpperCase()` returns `"HI MARY!"`.

- هذي الدالة تحول كل السلسلة النصية الى **حروف كبيرة capital** وهي من نوع **String**



String trim()

Returns a string with the same characters as the calling object string, but with leading and trailing white space removed. Whitespace characters are the characters that print as white space on paper, such as the blank (space) character, the tab character, and the new-line character '\n'.

EXAMPLE

After program executes `String pause = " Hmm ";`
`pause.trim()` returns "Hmm".

- هذي الدالة تسوي قص او حذف للمسافات الفارغة الموجودة في اطراف السلسلة النصية وهي من نوع **String**

char charAt(Position)

Returns the character in the calling object string at the *Position*. Positions are counted 0, 1, 2, etc.

EXAMPLE

After program executes `String greeting = "Hello!";`
`greeting.charAt(0)` returns 'H', and
`greeting.charAt(1)` returns 'e'.

- هذي الدالة تجيب لي الحرف من السلسلة النصية من خلال ال index الذي اضعه بين القوس وهي من نوع **char**



String substring(*Start*)

Returns the substring of the calling object string starting from *Start* through to the end of the calling object. Positions are counted 0, 1, 2, etc. Be sure to notice that the character at position *Start* is included in the value returned.

EXAMPLE

After program executes `String sample = "AbcdefG";`
`sample.substring(2)` returns "cdefG".

- هذي الدالة تسوي لي قص او قطع للسلسلة النصية من خلال ال index الذي اضعه بين القوس , وهي راج تقص السلسلة النصية من رقم ال index الى نهاية السلسلة النصية , وهي من نوع **String**

String substring(*Start*, *End*)

Returns the substring of the calling object string starting from position *Start* through, but not including, position *End* of the calling object. Positions are counted 0, 1, 2, etc. Be sure to notice that the character at position *Start* is included in the value returned, but the character at position *End* is not included.

EXAMPLE

After program executes `String sample = "AbcdefG";`
`sample.substring(2, 5)` returns "cde".

- هذي الدالة تسوي لي قص او قطع للسلسلة النصية من خلال ال index الاول وهو يعبر عن البداية و ال index الثاني وهو يعبر عن النهاية, وهي راج تقص السلسلة النصية من رقم ال index الأول الى رقم ال index الأخير, (حيث ال index الأخير لا يتم حسابه) , وهي من نوع **String**




```
int indexOf(A_String)
```

Returns the index (position) of the first occurrence of the string *A_String* in the calling object string. Positions are counted 0, 1, 2, etc. Returns -1 if *A_String* is not found.

EXAMPLE

After program executes `String greeting = "Hi Mary!";`
`greeting.indexOf("Mary")` returns 3, and
`greeting.indexOf("Sally")` returns -1.

- هذي الدالة تجيب لي رقم ال index لحرف او لبداية سلسلة نصية معينة , يعني اكتب بين القوسين اما حرف او string وهي راج تجيب رقم ال index , واذا كان الحرف او السلسلة النصية غير موجود في السلسلة النصية راج ترجع -1 يعني غير موجود , وهي من نوع **int**

❖ Escape Sequences

Display 1.6 Escape Sequences

```
\ " Double quote.  
\ ' Single quote.  
\ \ Backslash.  
\ n New line. Go to the beginning of the next line.  
\ r Carriage return. Go to the beginning of the current line.  
\ t Tab. White space up to the next tab stop.
```

- عشان اطبع رموز معينة او اتلاعب بالسلسلة النصية مثلا انزل سطر جديد اسوي مسافات راج استخدم Escape Sequences اكتب الشرطة المائلة \ ومن ثم الرمز المعين



❖ Comments:(التعليقات)

- **A line comment** begins with **double forward slash** symbol **//**
 - The compiler **ignore** this line
 - This type of comment is used for the code writer or for a programmer who modifies the code
- **A block comment** begins with forward **slash & star** symbols **/***, and ends with the symbol pair ***/**
 - The compiler ignores anything in between
 - This type of comment can span several lines
 - This type of comment provides documentation for the users of the program

- في جافا، هناك نوعان من التعليقات:
- **تعليقات السطر الواحد:** تبدأ برمزتين مائلين (**//**) وتستمر حتى نهاية السطر، يتجاهل compiler هذا السطر تمامًا. وتستخدم هذه التعليقات عادةً لشرح منطق بسيط أو تذكير المبرمج بشيء ما حول الكود.
- **تعليقات الكتلة:** تبدأ برمزتين (**/***) وتنتهي برمزتين (***/**). يمكن أن تمتد هذه التعليقات على عدة أسطر. تستخدم هذه التعليقات عادةً لتوفير توثيق شامل للمستخدمين أو المطورين

Display 1.8 Comments and a Named Constant

```
1  /**
2   Program to show interest on a sample account balance.
3   Author: Jane Q. Programmer.
4   E-mail Address: janeq@somemachine.etc.etc.
5   Last Changed: September 21, 2004.
6  */
7  public class ShowInterest
8  {
9      public static final double INTEREST_RATE = 2.5;
10
11      public static void main(String[] args)
12      {
13          double balance = 100;
14          double interest; //as a percent
15
16          interest = balance * (INTEREST_RATE/100.0);
17          System.out.println("On a balance of $" + balance);
18          System.out.println("you will earn interest of $"
19                          + interest);
20          System.out.println("All in just one short year.");
21      }
22  }
```

Although it would not be as clear, it is legal to place the definition of INTEREST_RATE here instead.

SAMPLE DIALOGUE

On a balance of \$100.0
you will earn interest of \$2.5
All in just one short year.



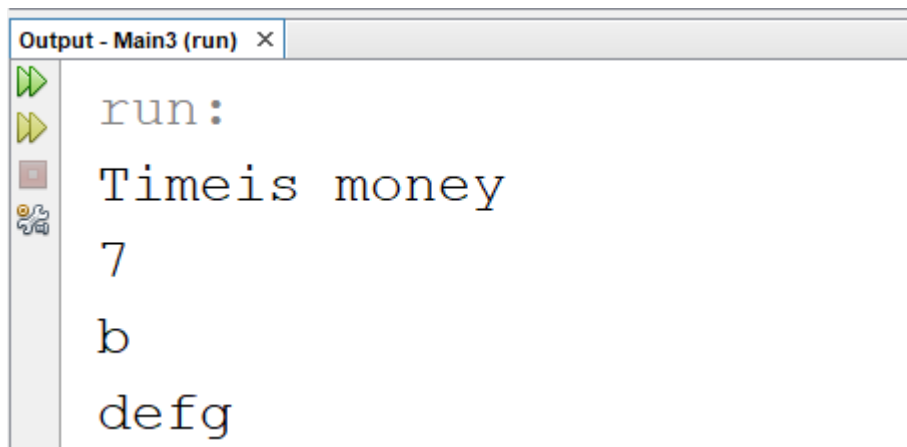
***What is the output produced by the following?**

```
String verbPhrase = "is money";  
System.out.println("Time" + verbPhrase);
```

```
String test = "abcdefg";  
System.out.println(test.length());  
System.out.println(test.charAt(1));
```

```
String test2 = "abcdefg";  
System.out.println(test2.substring(3));
```

Output



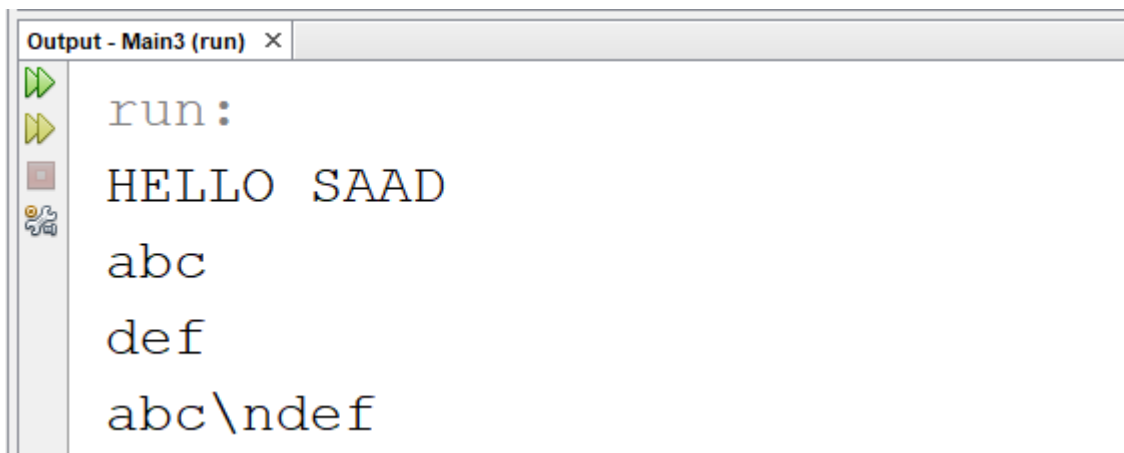
```
Output - Main3 (run) ×  
run:  
Timeis money  
7  
b  
defg
```



***What is the output produced by the following?**

```
String test = "Hello Saad";  
test = test.toUpperCase();  
System.out.println(test);  
  
System.out.println("abc\\ndef");  
  
System.out.println("abc\\ndef");
```

Output



The screenshot shows an IDE output window titled "Output - Main3 (run)". The output is as follows:

```
run :  
HELLO SAAD  
abc  
def  
abc\\ndef
```



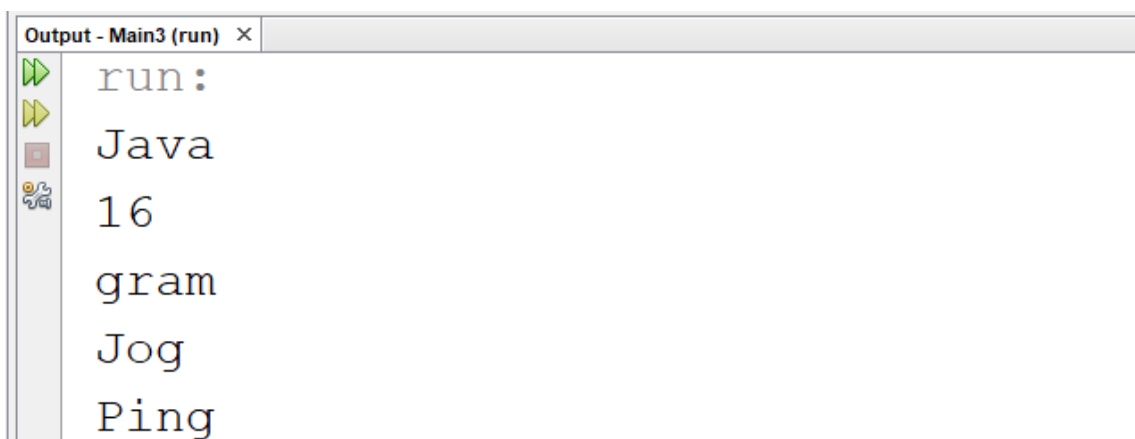
❖ Exercises:

What is the output

```
String text = "Java Programming";

System.out.println(text.substring(0, 4));
System.out.println(text.length( ));
System.out.println(text.substring(8, 12));
System.out.println(text.substring(0, 1) + text.substring(7, 9));
System.out.println(text.substring(5, 6) + text.substring(text.length() - 3, text.length()));
```

Output



```
Output - Main3 (run) x
run :
Java
16
gram
Jog
Ping
```

اونلاين

• تدرب على هذا الكود اونلاين من خلال الضغط هنا



053 359 7191



❖ Character Sets

- **ASCII:** A characters set used by many programming languages that contains all the characters normally used on an English-language keyboard, plus a few special characters
- Each character is represented by a particular number

• **ASCII:** مجموعة أحرف تستخدمها العديد من لغات البرمجة وتحتوي على جميع الأحرف المستخدمة عادةً على لوحة مفاتيح باللغة الإنجليزية، بالإضافة إلى بعض الأحرف الخاصة. و يتم تمثيل كل حرف برقم معين.

- **Unicode:** A characters set used by the java language that includes all the ASCII characters plus many of the characters used in languages with a different alphabet from English

• **Unicode:** مجموعة أحرف تستخدمها لغة جافا وتضم جميع أحرف ASCII بالإضافة إلى العديد من الأحرف المستخدمة في اللغات ذات الأبجديات المختلفة عن الإنجليزية مثل العربية.



❖ ASCII Encoding

Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20		100 0000	100	64	40	@	110 0000	140	96	60	`
010 0001	041	33	21	!	100 0001	101	65	41	A	110 0001	141	97	61	a
010 0010	042	34	22	"	100 0010	102	66	42	B	110 0010	142	98	62	b
010 0011	043	35	23	#	100 0011	103	67	43	C	110 0011	143	99	63	c
010 0100	044	36	24	\$	100 0100	104	68	44	D	110 0100	144	100	64	d
010 0101	045	37	25	%	100 0101	105	69	45	E	110 0101	145	101	65	e
010 0110	046	38	26	&	100 0110	106	70	46	F	110 0110	146	102	66	f
010 0111	047	39	27	'	100 0111	107	71	47	G	110 0111	147	103	67	g
010 1000	050	40	28	(100 1000	110	72	48	H	110 1000	150	104	68	h
010 1001	051	41	29)	100 1001	111	73	49	I	110 1001	151	105	69	i
010 1010	052	42	2A	*	100 1010	112	74	4A	J	110 1010	152	106	6A	j
010 1011	053	43	2B	+	100 1011	113	75	4B	K	110 1011	153	107	6B	k
010 1100	054	44	2C	,	100 1100	114	76	4C	L	110 1100	154	108	6C	l
010 1101	055	45	2D	-	100 1101	115	77	4D	M	110 1101	155	109	6D	m
010 1110	056	46	2E	.	100 1110	116	78	4E	N	110 1110	156	110	6E	n
010 1111	057	47	2F	/	100 1111	117	79	4F	O	110 1111	157	111	6F	o
011 0000	060	48	30	0	101 0000	120	80	50	P	111 0000	160	112	70	p
011 0001	061	49	31	1	101 0001	121	81	51	Q	111 0001	161	113	71	q
011 0010	062	50	32	2	101 0010	122	82	52	R	111 0010	162	114	72	r
011 0011	063	51	33	3	101 0011	123	83	53	S	111 0011	163	115	73	s
011 0100	064	52	34	4	101 0100	124	84	54	T	111 0100	164	116	74	t
011 0101	065	53	35	5	101 0101	125	85	55	U	111 0101	165	117	75	u
011 0110	066	54	36	6	101 0110	126	86	56	V	111 0110	166	118	76	v
011 0111	067	55	37	7	101 0111	127	87	57	W	111 0111	167	119	77	w
011 1000	070	56	38	8	101 1000	130	88	58	X	111 1000	170	120	78	x
011 1001	071	57	39	9	101 1001	131	89	59	Y	111 1001	171	121	79	y
011 1010	072	58	3A	:	101 1010	132	90	5A	Z	111 1010	172	122	7A	z
011 1011	073	59	3B	;	101 1011	133	91	5B	[111 1011	173	123	7B	{
011 1100	074	60	3C	<	101 1100	134	92	5C	\	111 1100	174	124	7C	
011 1101	075	61	3D	=	101 1101	135	93	5D]	111 1101	175	125	7D	~

❖ Numeral systems:

- Binary Numeral System - Base-2
- Octal Numeral System - Base-8
- Decimal Numeral System - Base-10
- Hexadecimal Numeral System - Base-16

- معنا بالأنظمة العديدة أربع أنواع : النظام الثنائي والثماني والعشري والسادس عشري وهي غير مهمة في هذا المقرر , مهمة في مقرر logical



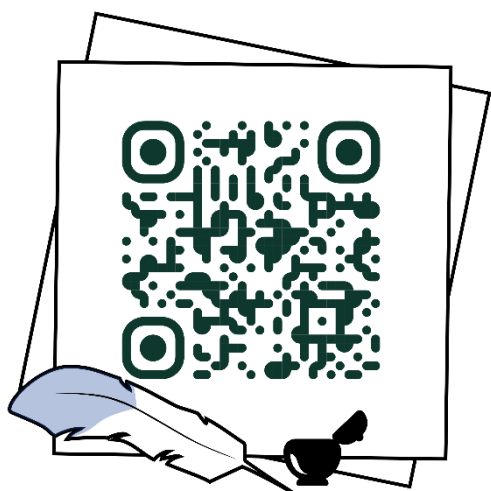
053 359 7191



ALZEEKA Tutorial

شروحات - مشاريع - خدمات - تصاميم

إنضم الآن



053 359 7191

