Welcome to the

Java Course

Module 1 – Day 04

Content of the course

- Introduction to programming
- Basic programming concepts
- Variables and Data Types
- Conditionals
- **Loops**
- **Control Structures**
- **Introduction to algorithms**

Day 1

Day 2

Day 3

.
.
.

Project Students - Step 3

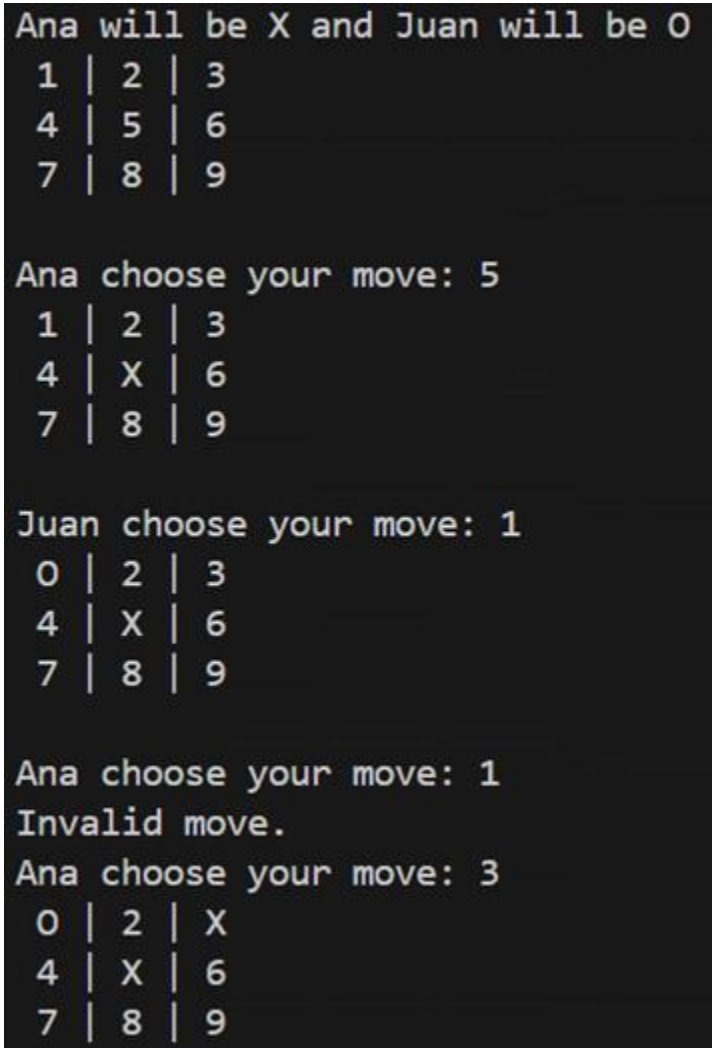
Modify the program such that it asks for the birth year until the user enters a valid one

```
Enter first name: Ana
Enter last name: Gaggero
Enter birthday (day of month): 22
Enter birth month: 10
Enter birth year: 2027
The birth year cannot be in the future, please enter a valid birth year: 2028
The birth year cannot be in the future, please enter a valid birth year: 1982
Enter course registered: Java

Student Name: Ana Gaggero
Date of Birth: 22 October 1982
Age: 41
Course Registered: Java
```

Project Tic Tac Toe

Step 3

- Add a loop to allow 9 moves.
- Add a variable to keep track of who's turn it is
- Check if the place is empty before updating the board. If it's not empty, print "Invalid move"



Let’s Recap

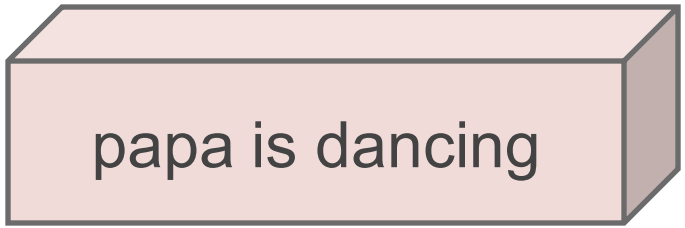
- While loop
- String operations
- Type casting

Recap String operations

- `str1 = str1.concat(str2);`
- `str.charAt(2);`
- `str.toLowerCase();`
- `str.toUpperCase();`
- `str1.equals(str2);`
- `str1.equalsIgnoreCase(str2);`
- `str1.contains(str2);`
- `str1.endsWith(str2);`
- `str1.startsWith(str2);`

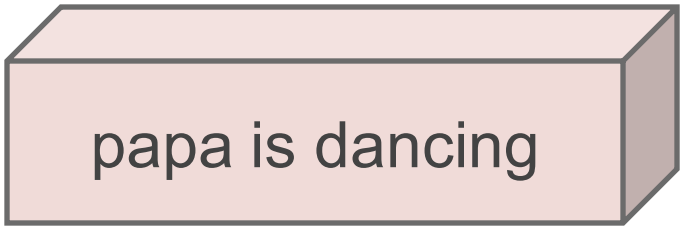
Recap String operations

```
String str = "Papa is dancing";

if (str.startsWith("p"))
  System.out.println(str.toLowerCase());
else
  System.out.println(str.toUpperCase());
```



Recap String operations

```
String str = "Papa is dancing";

if (str.startsWith("p"))
  System.out.println(str.toLowerCase());
else
  System.out.println(str.toUpperCase());
```



Recap String operations

```
String str = "Keep coding and never stop learning.";

if (str.charAt(5) == 'c’){
   System.out.print(str.charAt(0));
   System.out.print(str.charAt(6));
   System.out.print(str.charAt(7));
   System.out.print(str.charAt(17));
}
```

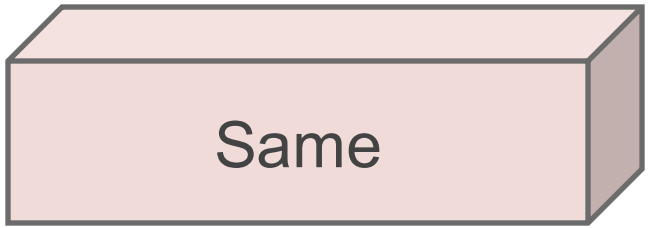
Recap String operations

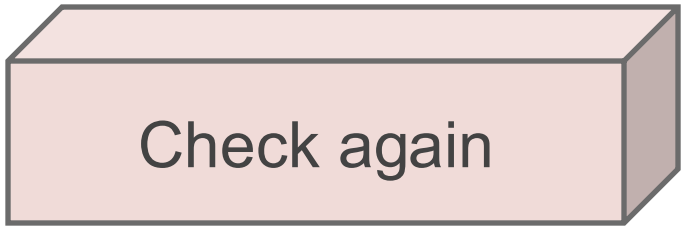
K [0]	e [1]	e [2]	p [3]	[4]	c [5]	o [6]	d [7]	i [8]	n [9]	g [10]

```
String str = "Keep coding and never stop learning.";

if (str.charAt(5) == 'c’){
   System.out.print(str.charAt(0));
   System.out.print(str.charAt(6));
   System.out.print(str.charAt(7));
   System.out.print(str.charAt(17));
}
```

Recap String operations

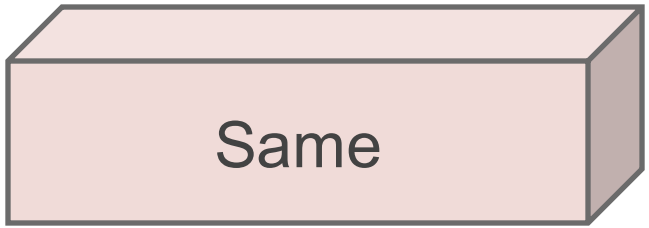
```
String str1 = "I would like a coffee please";
String str2 = "I would Like a coffee Please";

if (str1.equals(str2))
    System.out.println(”Same");
else
    System.out.println(”Check again");
```





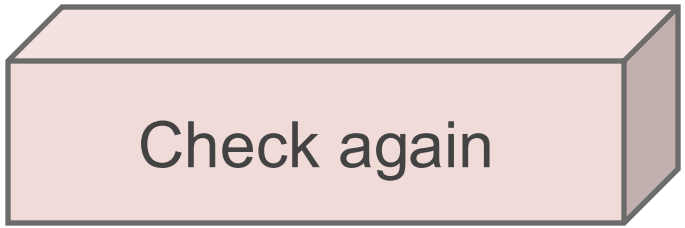
Recap String operations

```
String str1 = "I would like a coffee please";
String str2 = "I would Like a coffee Please";

if (str1.equals(str2))
    System.out.println(”Same");
else
    System.out.println(”Check again");
```



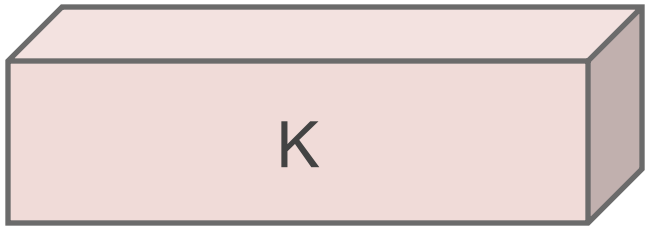
Recap String operations

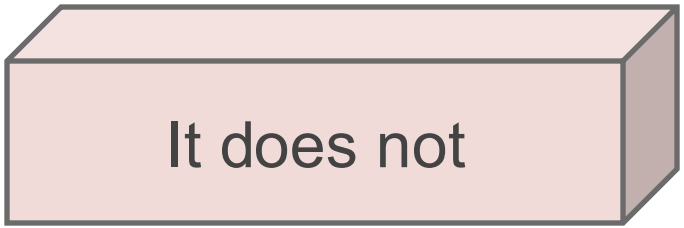
```
String str1 = "I would like a coffee please";
String str2 = "I would Like a coffee Please";

if (str1.equalsIgnoreCase(str2))
    System.out.println(”Same");
else
    System.out.println(”Check again");
```



Recap String operations

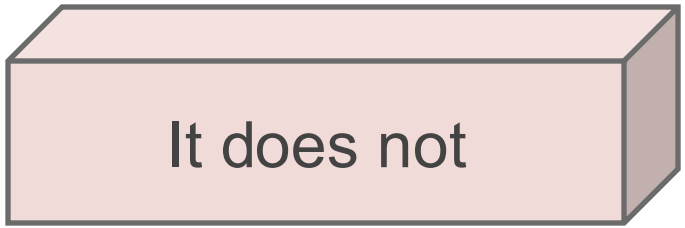
```
String str = "Keep coding and never stop learning.";

if (str.contains("ea")){
  System.out.print(str.charAt(0));
} else {
  System.out.println("It does not");
}
```





Recap String operations

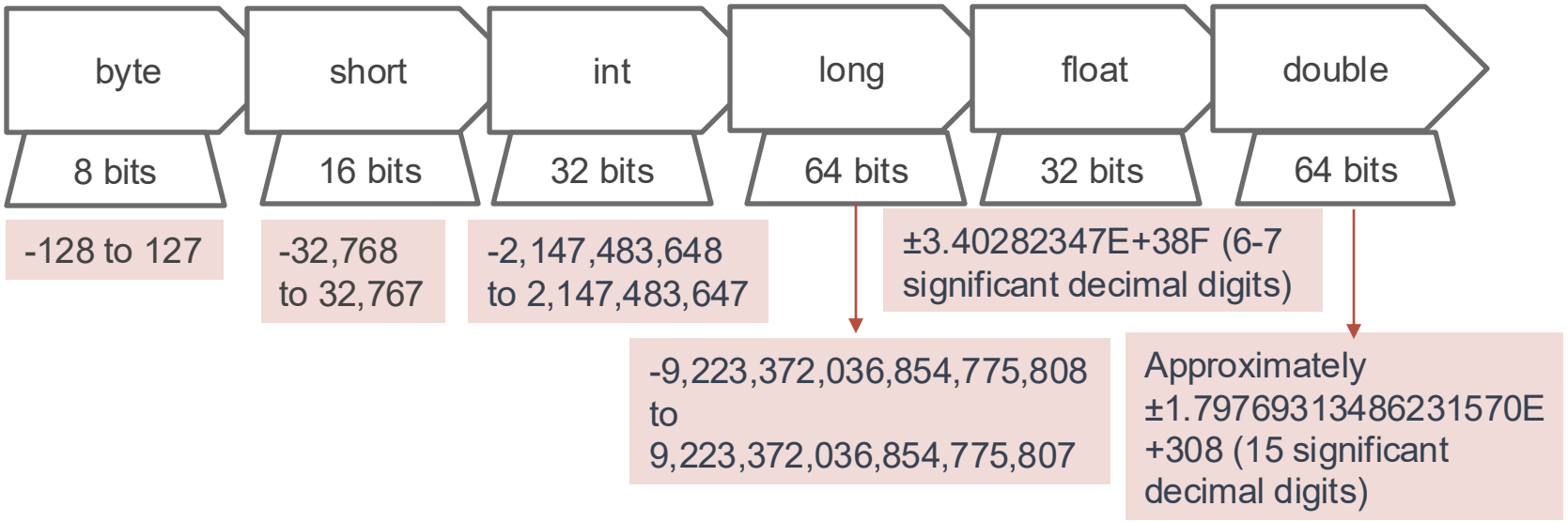
```
String str = "Keep coding and never stop learning.";

if (str.contains("ea")){
  System.out.print(str.charAt(0));
} else {
  System.out.println("It does not");
}
```



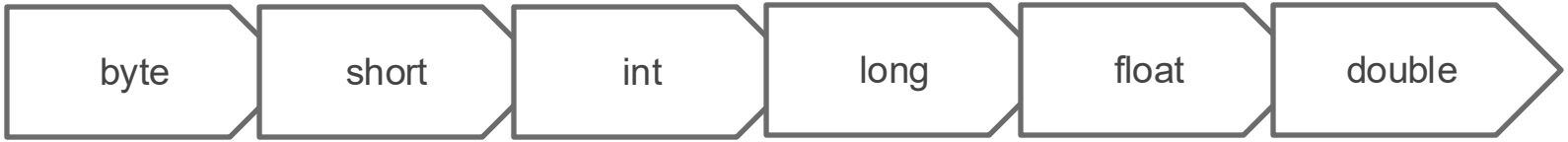
Type casting

Implicit & Explicit

```
1 int myInt = 9;
2 long myLong = myInt;
```

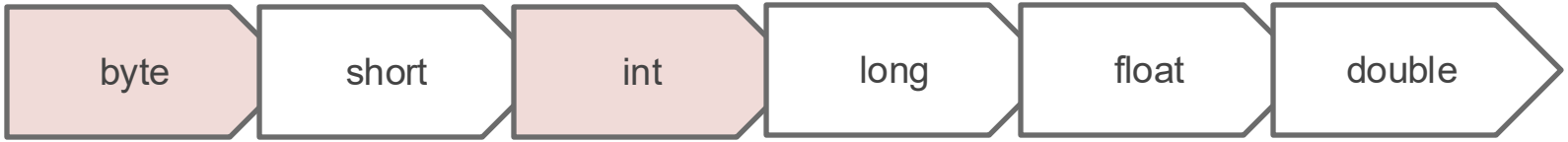


Type casting



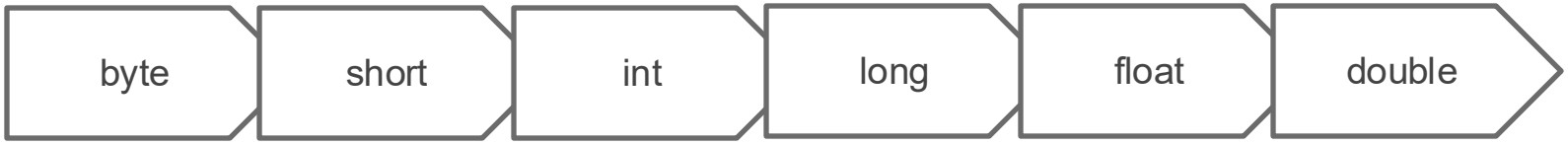

```
1 byte myByte = 100;
2 int myInt = myByte;
```

Type casting



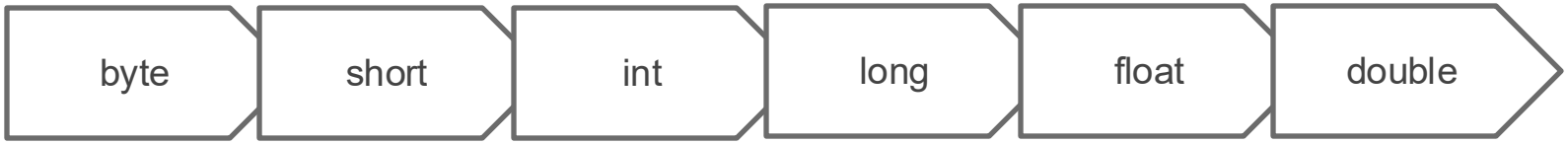

```
1 byte myByte = 100;
2 int myInt = myByte;
```

Type casting




```
1 double myDouble = 9.78;
2 int myInt = myDouble;
```

Type casting



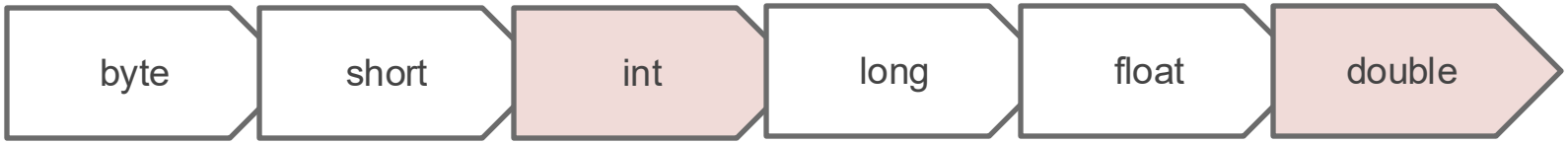

```
1 double myDouble = 9.78;
2 int myInt = myDouble;
```

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Type mismatch: cannot convert from double to int

Implicit

Explicit

Type casting




```
1 double myDouble = 9.78; // Outputs 9.78
2 int myInt = (int) myDouble; // Outputs 9
```

While Loop

When you don’t know how many times you have to perform a block of code, you need the while loop:

```
1 while (condition) {
2 // Code to be executed
3 }
```

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Let’s Learn

- Do-while
- For loop

Do-While Loop

If you want to execute the code in any case at least 1 time and then check the condition, you can use a do-while loop:

```
1 do{
2 // Code to be executed
3 }while (condition);
4
```

Loop

A loop is a sequence of actions repeated a known or unknown number of times.

For Loop

When you know how many times you have to perform a block of code, you need the for loop:

```
1 for(initialization;condition;update){
2   // Code to be executed
3 }
```

For Loop

```
1 for ( int i=0 ; i<5 ; i++) {
2   System.out.println(“Hello”);
3 }
```

For Loop

output:

Hello
Hello
Hello
Hello
Hello

Now YOUR TURN !

Let’s do exercises

Project Students - Step 4

Modify the program to allow the registration of more students. Then print the list of all students.

```
How many students do you want to register? 3
>>> Student 1 <<<
Enter first name: Ana
Enter last name: Gaggero
Enter birthday (day of month): 22
Enter birth month: 10
Enter birth year: 1982
Enter course registered: Java
>>> Student 2 <<<
Enter first name: Carol
Enter last name: Muller
Enter birthday (day of month): 12
...
```

Project Students - Step 4

```
...
>>> Student 3 <<<
Enter first name: Tom
Enter last name: Grass
Enter birthday (day of month): 7
Enter birth month: 1
Enter birth year: 1980
Enter course registered: Java

List of registered students:
Ana Gaggero, 22 October 1982, Java
Carol Muller, 12 April 1990, Python
Tom Grass, 7 January 1980, Java
```

Project Tic Tac Toe

Step 4

- Check if someone won. Print the winner and exit the game.
- Otherwise, print that it was a tie

```
Juan choose your move: 7
 X | 2 | 3
 O | X | 6
 O | 8 | 9

Ana choose your move: 9
 X | 2 | 3
 O | X | 6
 O | 8 | X

Ana won!
```