[LANGUAGE SPECIFICATION IN JAVA]

UNIVERSITY OF
KARACHI

COMPILER CONSTRUCTION 501
SUBMITED MISS ARISHA

GROUP MEMBERS

ALI UMAIR EB19103013
HARIS ALI EB19103033
M. ZANAEN ULLAH EB19103089

Table of Contents

DATA TYPES

Keywords	Class Part	Value Part	Syntax
digit	Datatype	int	digit myName=33;
point	Datatype	float	point myPointNum=6.99;
char	Datatype	char	char myChar="a";
text	Datatype	String	Text myText="hello world"
bool	Datatype	boolean	bool myBool=True;

Note:-

- Here we use digit instead of int .
- Here we use point instead of float.
- Here we use text instead of String.

PUNCTUATORS

Punctuators	Class Part	Value Part
;	;	;
:	:	:
.	.	.
,	,	,
(	(	(
)	)	)
{	{	{
}	}	}
[	[	[
]	]	]

OPERATORS

Punctuators	Class Part	Value Part
+	Arithmetic Operators	+
-	Arithmetic Operators	-
*	Arithmetic Operators	*
/	Arithmetic Operators	/
%	Arithmetic Operators	%
==	Relational Operators	==
!=	Relational Operators	!=
>	Relational Operators	>
<	Relational Operators	<
>=	Relational Operators	>=
<=	Relational Operators	<=
&&	Logical Operators	&&
||	Logical Operators	||
++	Unary Operator	++
--	Unary Operator	--
=	Assignment Operator	=

Example:-

```
class Main {
        general static void main() {
                digit a = 12, b = 5;

                System.out.println("a + b = " + (a + b));
                System.out.println("a – b  = " + (a - b));
                System.out.println("a * b   = " + (a * b));
                System.out.println("a / b   = " + (a / b));
                System.out.println("a % b  = " + (a % b));
        }
}
```

CONDITIONAL

Keywords	Class Part	Value Part	Syntax
if	Condition	if	If(/**condition*/){}
else	Condition	else	If(/**condition*/){} else{}

Example:-

```
class Main {
        general static void main() {
                digit a =0;

                if(digit > 0){
                 System.out.println("the number is positive.");
                }

                else if(digit < 0){
                 System.out.println("the number is
                negative."); }

                else{
                System.out.println("the number is Zero.")
                }
        }
}
```

LOOPS

Keywords	Class Part	Value Part	Syntax
for	loop	for	for(digit i=1:i<=5;i++){}
till	loop	while	till (i < 5){}
do	loop	do	do(i++) till(I <= 10)

Note:-

- Here we use till instead of while loop.

Example:-

```
class Main {
        general static void main() {
                digit a =5;
                loop ( digit i = 1; i <= n; i++ ){

                        System.out.println(i)

                        }

                }

}
```

ARRAY

- *One dimensional:-*

  Example:-

  ```
  digit [] array_01 = { 1 ,2 ,3 ,4 ,5 } ;
  ```

- *Multi dimensional:-*

  Example:-

  ```
  digit [][] array_02 = { { 1 ,2 ,3 ,4 ,5 } , { 6 ,7 ,8 ,9 ,0 } ) ;
  ```

STATEMENT

Keywords	Class Part	Value Part	Syntax
Case	*Statement*	*case*	*case 7:*
shift	*Statement*	*switch*	*Shift(digit){* *case 6:* *-----* *}*
break	*Statement*	*break*	*Case 7:* *----* ***break;***
continue	*Statement*	*continue*	*for(digit i=1:i<=5;i++){* *If(/**condition*/){* ***Continue;*** *}* *}*

ACCESS MODIFIERS

Keywords	Class Part	Value Part	Syntax
personal	*Access Modifier*	*private*	*personal class A{}*
general	*Access Modifier*	*public*	*general class A()*
protected	*Access Modifier*	*protected*	*protected class A()*

Note:-

- *Here we use personal instead of private.*
- *Here we use general instead of public.*

Example:-

```
class Animal {
        protected void display(){

                        System.out.println(" i am an animal ");

                }

        }
class dog extend Animal {
        general static void main() {
                Dog dog = new Dog();
        }
}
```

OOPS CONCEPTS

- *Inheritance*
- *Abstraction*
- *Class, constructor, method, object*

Keywords	Class Part	Value Part	Syntax
Class	*OOPs Concept*	*Class*	*class A{}*
extend	*OOPs Concept*	*extend*	*class A()extend class b()*
static	*Reference*	*static*	*static text name="hello"*
super	*Reference*	*super*	*super./*method*/()*
this	*Reference*	*this*	*this./*method*/()*
new	*OOPs Concept*	*new*	*Class A obj = new class B()*
Implements	*OOPs Concept*	*Implements*	*class A() implement class b()*

Example:-

```
class Main {
        digit I;
        personal Main(){

                I=5;

                System.out.println(" constructor is called");

                }

                general static void main() { Main obj =
                new Main(); System.out.println(" value of
                1 : " + obj.i");

                }

        }

interface Animal {
        protected void display(){

                        System.out.println(" i am an animal ");

                }

        }
class dog implement Animal {
        general static void main() {
                Dog dog = new Dog();
        }
}
```