

Lecture 1 Programming

Program 1(Input/Output)

```
1 // Program #1 input,output [This is a comment not a code we use // for line comments ]
2 // Written by : Mohamed Ibrahim
3
4 import java.util.*; // We import this package to enable us to use "Scanner" class to receive data from user
5 public class ProgramOne { // We MUST name the PUBLIC class as same as the name of the JAVA file
6
7     public static void main(String[] args) //Every program should have a main just for now type this line as it
8     {
9         Scanner objectname = new Scanner (System.in); /* This line to make object to receive
10            input ** objectname can be changed to any name */
11
12         String input ; // A variable of type string note that 'S' is capital letter
13
14         int number;
15
16         System.out.print("Enter your name : "); //This is the function used to print like "COUT" in c++
17
18         input = objectname.nextLine(); /*we store what is received in the variable
19            input -- ".nextLine" used to receive String*/
20
21         System.out.println("Welcome"+" "+input+" !"); // "+" is used to combine variables with strings
22
23         System.out.print("Enter any number : ");
24
25         number = objectname.nextInt(); //nextInt() used to receive integer input then be stored in the var. number
26
27         System.out.print("Your number is " + number);
28     }
29 }
30 }
```

Read the comments in the code above

Data types

- int – takes 4 bytes of memory (32 bit) – store integers
- long – takes 8 bytes of memory (64 bit) – store integers
- float – takes 4 bytes of memory (32 bit) – store numbers with float point
- double – takes 8 bytes of memory (64 bit) – store numbers with float point
- char – takes 2 bytes of memory (16 bit) – store characters with ASCII code

ASCII Code(American Standard Code for Information Interchange)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

Difference between (char x =5 ;) and (char x ='5' ;)

Assuming that char takes 8 bits only so char x = 5 ; will be like that x : 00000101 which is 5 in DEC

Assuming that char takes 8 bits only so char x ='5' ; will be like that x : 00110101 which is 53 in DEC

So we conclude that if we add the ' ' the decimal ASCII code value will be stored in the character

if not the value given will be stored as it is but limited to 255

Note that char x ='48' will be **Compilation error** as 48 is *not a character*

Difference between (char x =50 ;) and (int y =50 ;)

Actually there is no difference in storing except the **size** of each one but there is a difference while we **print** them in the console :-

If we printed **x** we will get **2** as **50** is the DEC ASCII code of the **character 2**
but If we printed **y** we will get **50** as we wrote .

Casting

Casting is a method to covert between different data types.

so if we have **char x ='8' ; int y =10 ;** we can write **y=x;** then **56** will be assigned to **y**
as **56** is the ASCII code of **'8'**

but if we want the opposite we have to use casting so we write **x = (char)y;**

Note that **String** can not be converted to **int** in this way [Search for it]

Conditions

Ternary Operator (? :)

It takes the form **<condition> ? <expreion1> : <expreion2> ;**

if the condition is **true** expreion1 is returned **else** expreion2 is returned for example

```
int v = 90;  
String result = (v>50) ? "Passed" : "Failed" ;
```

we check if **v larger than 90** the String **result** will be Passed else will be Failed

if else statement

It takes the form **if** (<condition>) {<statements>} **else** {<statements>}

```
int v = 90;  
String result ;  
if(v>50) { result ="Passed";} else {result = "Failed";}
```

Note that we can add **more than 1** condition inside the **if**

```
if(v>50 && v<100) { result ="Passed";} else {result = "Failed";}
```

Program 2 (Passing the exam) solution 1

```
1 import java.util.*;  
2 public class Test  
3 {  
4     public static void main(String args[])  
5     {  
6         Scanner input = new Scanner(System.in);  
7         int grade;  
8         String result="";  
9         System.out.print("Enter your Grade :");  
10        grade = input.nextInt();  
11  
12        if(grade < 50) {result = "Not Passed";}   
13        if(grade >= 50) {result = " Passed";}   
14        if(grade >= 65) {result = "Good";}   
15        if(grade >= 75) {result = "Very good";}   
16        if(grade >= 85) {result = "Excellent";}   
17        System.out.print("You "+result);  
18  
19    }  
20  
21 }
```


Program 2 (Passing the exam) solution 2

```
1 import java.util.*;
2 public class Test
3 {
4     public static void main(String args[])
5     {
6         Scanner input = new Scanner(System.in);
7         int grade;
8         String result="";
9         System.out.print("Enter your Grade :");
10        grade = input.nextInt();
11
12        if(grade < 50) {result = "Not Passed";}
13        else if(grade >= 50) {result = " Passed";}
14        else if(grade >= 65) {result = "Good";}
15        else if(grade >= 75) {result = "Very good";}
16        else if(grade >= 85) {result = "Excellent";}
17        System.out.print("You "+result);
18    }
19 }
20
21 }
```

Solution 1 vs Solution 2

Solution 1 is more readable than 2 (Quality,Readability) as it checks every **if** statement,Solution 2 is faster(Performance)

Dr.mahmoud says that solution 1 is better than 2 as it easy to maintain be other developers.

Basic Microsoft interview question :

Write a program to define Triangle Type by its sides length given from input !

Common **error** to equal 3 var. as **(x==y==z)** it should be like that **(x==y&& y==z)**

```
1 import java.util.*;
2 public class Triangle
3 {
4     public static void main(String[] args) //Every program should have a main just for now type this line as it
5     {
6         int sidel,side2,side3;
7         Scanner getSide = new Scanner(System.in);
8         System.out.print("Enter Sidel : ");
9         sidel = getSide.nextInt();
10        System.out.print("Enter Side2 : ");
11        side2 = getSide.nextInt();
12        System.out.print("Enter Side3 : ");
13        side3 = getSide.nextInt();
14
15        if(sidel==side2 && side2 == side3 ) {System.out.print("This is Equilateral");}
16        else if(sidel==side2 || side2 == side3 || sidel == side3 ) {System.out.print("This is Isosceles");}
17        else {System.out.print("This is Scalene");}
18    }
19 }
20
21 }
```