MOKSHITH

WELCOME
TO
MY

JAVA BOOK

LEARN JAVA HERE

1) Write a Java program to accept two variables and find the difference.

calling ... basic package

```
import java.util*; //importing a package
public class test // creating a class
{
 void method()//[ to start a program][void = 0]
 {
 int a,b,c;//[int = integer(numbers only)]
 Scanner Sc=new Scanner(System.in);
 a=Sc.nextInt();//[ for accepting them
 b=Sc.nextInt();  [ others are given in upcoming ]//
 c=a-b; // [declaring]
 System.out.println("The sum of 2 nos=" +c);
 }
}
```

Variable description

variable	data type	function
a	int	for storing 1st number
b	int	for storing 2nd number
c	int	for storing the difference

Scanner statement

meaning:- It reads the data from the user

data type

Int a;	a = Sc.next Int(),	[numbers without decimals]
double a;	a= Sc.next Double();	[numbers with decimals]
Float a.	a = Sc.next Float();	[numbers with decimals]
char a;	a= Sc next ().Char AT(0);	[characters]
long a;	a = Sc.next long();	[numbers without decimals]
string a;	a = Sc.next();	[for one word]
String a;	a = Sc.next Line(	for multiple words]

OPERATORS

basic [arithmetic operators]

+ [addition]
- [subtraction]
* [multiplication]
/ [division (returns quotient)]
% [division (returns remainder)] [known as modulus]

Relational operators [used to compare two values]

> greater than
< less than
== equal to
!= not equal to
>= Greater than or equal to
<= less than or equal to
* returns boolean

Primitive datatype
Int, double, float, char
byte, short, long
and boolean
Non Primitive datatype
Classes
Arrays
Interface

Logical operators

&&; And; it returns true when both the condition are true
||; Or; " " " " " one of the condition is true
!; NOT; it returns true when expression is false. ex: !(5>3)=T

assignmment operator - =

shorthand operators

Expression	shorthand
a= a+b	a+=b
a= a-b	a-=b
a= a*b	a*=6
a=a/4	a/=4
a= a%3	a%=3

$$\begin{tikzcd}
& \text{Increment and Decrement} \arrow[dl, dash] \arrow[dr, dash] & \\
\text{Prefix} \arrow[d, dash] & & \text{Postfix} \arrow[d, dash] \\
++X \text{ or } --x & & x++ \text{ or } X--
\end{tikzcd}$$

Prefix:- When you use the increment operator as a prefix, such as ++j, the value of j is increased by 1, and then it returns the value.

postfix:- When you use the increment operator as a postfix, such as j++, the orginal value of j is returned first, and then j is incremented by 1

The decrement operator's prefix and postfix work similarly to the increment operator's prefix and postfix, except, the decrement decreases by 1

Ternary operator

The Ternary operator is also known as conditional operator. It works on three or more operands. This operator is used to decide for a value based on given conditions. It is a condensed of if-else statement that also returns a value.

Syntax :-

```
    Variable = (Boolean Expression) ? Ex2 : Ex3;
[Ex:- Expression]
```

In ternary construct, the first expression must be a Boolean. It returns expression 2 as an output if Expression 1 evaluates to true otherwise, It returns Expression 3.

Q) WAJP to find greater number between 2 numbers using ternary operator[?]

```
import java.util.*;
public class Ternary
{
 void method()
 {
 int n1=10,n2=40,max;
 System.out.println("First number:" +n1);
 System.out.println("Second number:" +n2);
 max =(n1>n2) ? n1:n2 ;
System.out.println("maximum is =" + max);
  }
 }
```

output

```
First number:10
 Secondnumber:40
     Maximum is:40
```

Variable description

Data type	Variable	function
n1 Int	n1	To store the value of 1no
n2 Int	n2	To store the value of 2no
max Int	max	To store the value of max

Conditional Constructs in Java

Decision making statements

if
if - else
if - else - if
switch

If and its versions

The if statement is the simplest decision-making statement. It executes a statement or block of statements associated with it if a test expression (condition) evaluates to true. otherwise, the flow of control comes out of the if block and executes the next statement

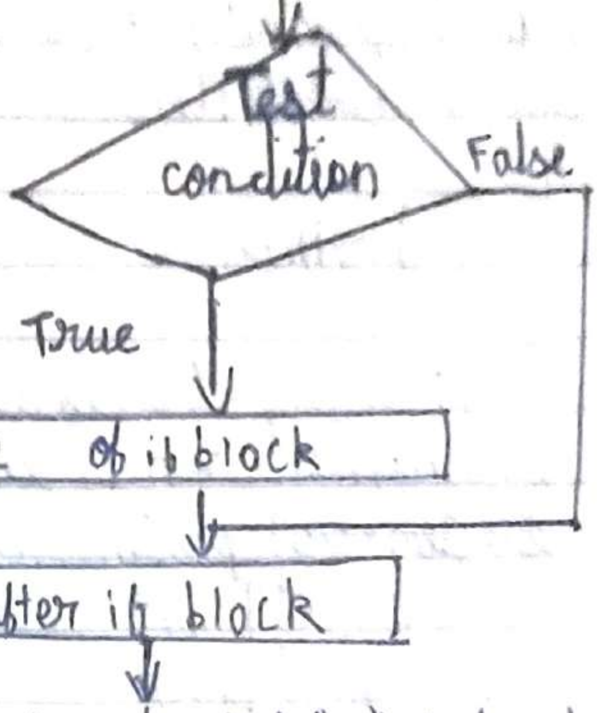
The if-else statement is an extended version of the if statement.
If the if condition is false, it goes to the else block and statements associated with it gets executed.

If there are multiple conditions to decide an action to be performed, This type of structure is known as the if-else ladder.
The first condition is "If"
The rest all are else If
The last one is else [If none of the condition are true, it goes

If

Syntax :-
```
if (condition)
statement;
```

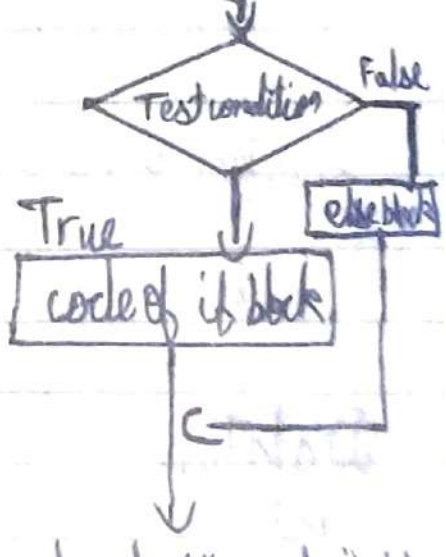
Example
```
int m;
m = Sc.nextInt();
if (m>40)
System.out.println("pass");
```



Flow-chart of "if" statement

If - else

Syntax:
```
  if(condition)
  statement 1;
  else
  statement 2;
```

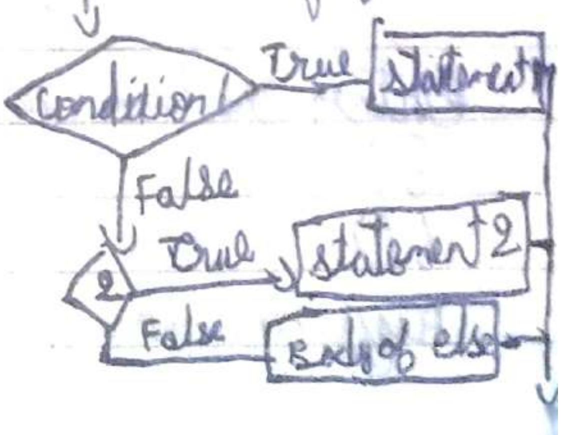


Flow-chart of "if-else" statem

If-else-if

Syntax:-
```
if (condition 1)
 statement 1;
else if (condition 2)
 statement 2;

 ...
else
statement;
```



9) WASP that accepts 2 numbers and prints the larger of the 2.

```
import java.util.*; // importing a package
public class ifelse // creating a class
{
  void method()
 {
  int a,b; // declaring variables
 Scanner Sc = new Scanner(System.in);
 System.out.println(" enter the two numbers ");
  a = Sc.nextInput;
  b = Sc.nextInput;
  if (a >b)
 System.out.println(" Bigger number =" + a);
  else
 System.out.println(" Bigger number =" +b);
   }
   }
```

VD

Datatype	Variable	Function
int	a	For storing value of a
int	b	For storing value of b

Output

```
enter the two numbers
57
60
Biggernumber = 60
```

Q) WAJP to input the marks obtained by a student and print his grade according to the conditions:

- Marks greater than or equal to 90, grad A
- Marks greater than or equal to 75 but less than 90, Grade B
- Marks greater than or equal to 60 but less than 75, Grade C
- Marks greater than or equal to 4 but less than 60, Grade D
- Marks less than 40, Grade E

```
import java.util.*;
public class marks
{
void method()
{
int m;
Scanner Sc = new Scanner(System.in);
System.out.println("Enter marks");
m= Sc.nextInt();
if (m>=90)
  System.out.println("your grade is A");
else if(m>=75 && m<90)
  System.out.println("your grade is B");
else if ( m>=60 && m<75)
  System.out.println("your grade is C");
else if ("m>=40 && m<60)
    System.out.println("D");
else
  System.out.println("E");
}
}
```

variable	data type	vD	Function
m	int		to store the marks

Output:-

```
Enter marks
 85
Your grade is B
```

Switch-case

To overcome the problems of using multiple if statements, you can use the switch & case statement This Statement is used in menu-driven programs where the user has to select an option from given multiple statements The switch statement is a multiway branch statement. It provides an easy way to exacute the diffrent parts of the code based on the value of the expression.

Syntax:

```
 switch(expression)
 {
 case 1:
 statement 1;
 break;
 case 2:
 statement;
 break
```

default:
statement;
}
The break statement takes the control out of the switch-case. if the condition of the switch statement does not match with any of the cases, the default statement gets executed

Q) WAJP to input the day no and print the name of the day using switch case.

```
import java.util.*;  // using a package
public class switch
{
 void method ()
 {
 int d;                    //declaring variables
 Scanner Sc = new Scanner(System.in);
 System.out.println("Enter day number");
 d = Sc.nextInt(); //accepting value from user
 switch (d)   // initialing switch-case construct
 {
  case 1:
      System.out.println("Monday");
      break;
  case 2
System.out.println("Tuesday");
break;
case 3:
System.out.println("Wednesday");
break;
case 4:
System.out.println("Thursday");
break;
case 5:
System.out.println("Friday");
break;
case 6:
System.out.println("Saturday");
break;
case 7:
System.out.println("Sunday");
break;
default:
System.out.println("Invalid day no");
}
}
}
```

Variable-description

Datatype	variable	function
int	d	storing value of day from the user

Q) WAJP to input the month no and tell the no of day in it.

```
import java.util.*;    //using util package
public class month
{
  Scanner Sc=new Scanner(System.in);
  int m;        //declaring variables
  System.out.println("Enter month number:");
  m = Sc.nextInt();//accepting from user
  switch(m)  //initiating switch construct
  {
   case1:case3:case5:case7:case8:case10:case12:
   System.out.println("31 days");
   break;
  case4:case6:case9:case11:
   System.out.println("30 days");
   break;
   case2:
   System.out.println("28days");
   break;
   default:
   System.out.println("Invalid choice");
   }
   }
   }
```

Menu - driven programs
Electric - Bill

Q) WAJP to input the unites of electricity consumed and find the bill charged at the following rates

Units	charges
First 100 units	₹2 per unit
Next 100 units	₹3 per unit
Next 200 units	₹4 per unit
Above 400 units	₹5 per unit

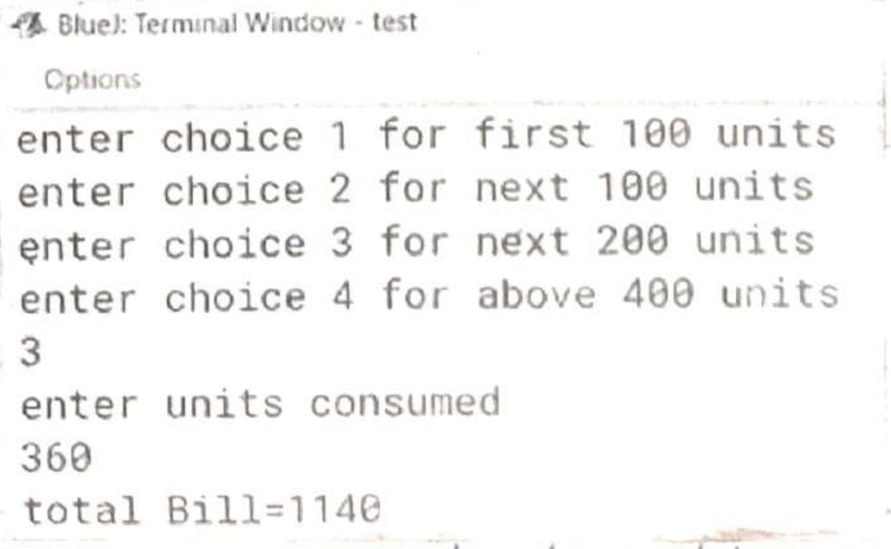
```
import java.util.*;
public class bill
{
void method ()
{
int u,b,ch;
Scanner Sc = new Scanner (System.in);
System.out.println ("Enter choice 1 for first 100 units");
System.out.println ("Enter choice 2 for next 100 units");
System.out.println("Enter choice 3 for 200 units");
System.out.printlN("Enter choice 4 for above 400 units");
  ch = input.next Int ();
 switch (ch)
  {
   case 1:
          System.out println (" Enter units consumed");
```

```
u = input.next Int();
b = u*2;
System.out.println("Total Bill =" +b);
break;
case 2:
    System.out.println ("Enter units consumed:");
    u=input.next Int();
    b=200+(u -100)*3;
   System.out.println("Total Bill =" +b);
break;
case 3:
   System.out.println("Enter units consumed");
  u=input.next Int ();
  b=500 + (u-200)*4;
  ~~System.out.println("Enter units consumed");~~
  ~~u= input.next Int()~~;
  b= System.out.println(" Total bill=" b);
      break;
case 4:
    System.out.println ("Enter units consumed");
   u=input.next Int();
   b=500+800 +(u-400)*5;
  System.out.println("Total Bill =" +b);
  default:
  System.out.println("Please enter avalid choice:");
System.exit(0);
}
}
}
```

Variable description

Datatype	Variable	function
int	u	to store the no of units consumed
int	b	to store the value of total bill
int	ch	to store the value of user input

out.Put:-


BlueJ: Terminal Window - test

Options

```
enter choice 1 for first 100 units
enter choice 2 for next 100 units
enter choice 3 for next 200 units
enter choice 4 for above 400 units
3
enter units consumed
360
total Bill=1140
```

Few points of switch case

+ The expression used in switch-case must be an integer or a character. Also, in case of character, only single-character is allowed. In JDK7, the expression can ~~only~~ also be a type string.
* Duplicate case values are not allowed.
+ The default statement is optional.

You can terminate a program by using the exit() method of the system class.
syntax :-
System.exit(n); [if n=0 normal means it does not check remaining statement. if n ≠ 0, abnormal]

* The break statement is used inside the switch to terminate a statement sequence.
* The break statement is optional. if omitted, the execution of the next case will continue. This is known as fall through.

Fall-through Situation

The term "fall through" refers to the switch statement executes the way its various case sections.

Every statement that follows the selected case section will be executed unless a break statement is encountered.

Example :-

Input:-
```
switch (Num)
{
 case 1:
 System.out.println("One");
 case 2:
 System.out.println("two");
  case 3:
```

```
System.out.println("three");
default:
System.out.println("Invalid");
}
```

Output

If the value of Num is 1 the output will be:

```
One
two
three
Invalid
```

if else V/s switch - case

if-else	switch-case
• The if statement is used to select between two alternatives.	• The switch statement is used to select among multiple alternatives.
• it can be used for a range of value	• Only for a single value
• The values used in the if statement are based on constraits.	• The expression used in switch can have values based on user's choice.
• Any data type can be used in the if-condition	• The expression used in switch-case must be an integer or a character, also • in case of a character, only single character is allowed

End of Conditional constructs in Java

Mathematical Library Methods

i) Math.sqrt()
It returns the square root of a positive no. It gives NaN when negative number is passed
Syntax :- Math.sqrt(n);
Ex i :- Math.sqrt(36)
output = 6.0
Ex ii :- Math.sqrt(-1)
output = NaN

ii) Math.cbrt();
This function returns the cube root of an integer. It returns a double value.
Syntax :- ~~Math~~ Math.cbrt(n);
Ex :- Math.cbrt(125) = 5.0

iii) Math.pow();
This function returns the pow of a variable raised to the other. It returns a double type value.
Syntax :- Math.pow(a,b);
Ex :- Math.pow(2,3) = 8.0

iv) Math.max():
This function is used to find the bigger of two numbers. Its return type

will depend upon the type of values used in the function.
Syntax: Math.max(a,b);
Example:- Math.max(15,45) = 45
Math.max(7.5,14.5) = 14.5

v) Math.min():
The function returns the minimum of two numbers. It's return type will depend upon the type of values used in the function.
Syntax:- Math.min(a,b);
Example:- Math.min(15,45) = 15
Math.min(7.5,14.5) = 7.5

vi) Math.abs():
This function returns the absolute value of an argument without its sign.
Syntax:- Math.abs(a);
Example:- Math.abs(-12.45); = 12.45
Math.abs(10); = 10

vii) Math.round():
This function returns the rounded value of a number given within the brackets. Its return type is Integer.

Syntax :- Math.round(a);
Example :- Math.round(2.4); :- 2
Math round(-2.4); :- -2

viii) Math.ceil():
This function returns the next higher integer value of a number given within the brackets.
Syntax :- Math.ceil(a);
Example :- Math.ceil(10.4); :- 11
Math.ceil(10.7); :- 11
Math.ceil(-15.10); :- -15
Math ceil(-10.5) :- -10.

ix) Math.floor();
This function returns the previous lower integer value of a number given within the brackets. It's return type double.
Syntax :- Math.floor(a).
Example :- Math.floor(10.4): 10
Math.floor(10.7) : 10
Math.floor(-15.10) :- -16
Math.floor(-10.5) :- -15

x) Math.random():
It is used to generate ~~no~~ no greater than or equal

to 0.0 and less than 1.0

If you want to generate between 1 and n.
int r = (int)(Math.random() * n)+1; where n is the upper limit

Syntax :- d = Math.random();

X.) Math.rint():
This function returns the nearest even integer value of a given fractional number. The return datatype will always be double.
Example:- Math.rint(8.5); = 8.0
Math.rint(7.5); = 8.0

Math.rint(-9.5) = -10.

XI) Math.log():
This function returns the logarithmic value of the data given within the brackets. It results in a double type value. The base of the log is taken to 10.
Example:- Math.log(100) = 2.0