VARIABLES

(primitive)Variables – byte, int, long, float, char, Boolean

(non-primitive) String name = new String(original:“abir”);

Sout (name.length());

String concatenate ( var + var1)

name.charAt(0)

name.length()

name.replace(old char:‘a’,new char:’b’) //the main string do not change. So, have to store in a different variable

name.substring(0,3)

ARRAY

Int[] marks = new int[3];

marks[0]=33;

Sout(marks[0]);

Sout(marks.length);

Arrays.sort(marks); //but have to include package > import java.util.Arrays;

Int[] marks = {23,55,67};

2D array : int[] [] finMark = {{1,2,3},{4,5,6}};

Sout (finMark[1] [1]); //result = 5

CASTING (change type)

Implicit casting happens by default :

Double price = 100.04;

Double finpr = price = 18;

Here double is 8 bytes and int is 4 bytes. So we can put int in double. But have to use explicit casting to put double into int

Explicit casting

Int p = 100;

Int fp = p + (int) 14.09;

CONSTANTS (cant change values)

Final float pi =3.14F;

OPERATORS

Plus, minus, multiply, div, modulo

Sout(A++); (first print then change value)

Sout(++A); (first change value then print)

Same for A—

MATH

Sout(Math.max(5,6)); //6

Sout(Math.min(5,6)); //5

Sout(Math.random()); //random values with long

Sout((int)(Math.random()\*100)); //random values inside 100

INPUT

Scanner sc = new Scanner (System.in); // have to import > import java.util.Scanner;

Sout (“input your age”);

Int age = sc.nextInt(); // for any primitive variables

String name = sc.next(); //for strings or word

String name = sc.nextLine(); //for strings or whole line

Comparison Operators :

A == b

A != b

A < b

A > b

A <= b

A >= b

CONDITIONAL STATEMENTS :  
Boolean isup = true;

If (isup == true)

Sout (“day”);

Else

Sout (“night”);

Logical operators - &&(and), ||(or), if( !isup ) – it is the opposite

SWITCH STATEMENT :

Int day = 1;

Switch (day) {

Case 1 :

Sout (‘a’);

Case 2 :

Sout (‘b’);

default :

Sout (‘c’); }

For day = 1 , Output = a,b,c

For day = 2 , Output = b,c

To get single output, we have to use break –

Int day = 1;

Switch (day) {

Case 1 :

Sout (‘a’);

Break:

Case 2 :

Sout (‘b’);

Break;

default :

Sout (‘c’); }

For day = 1 , Output = a

For day = 2 , Output = b

LOOPS :

For –

for (int i = 1; i<=100; i = i+1){

Sout(i);

}

While –

Int i = 100;

While(i >= 1){

Sout (i);

i = i – 1;

}

Do While (can be used as infinite loop )–

Int k = 100;

Do {

Sout(k);

k=k-1;

} while(k>=1);

Infinite loop –

Scanner sc = new Scanner(System.in);

Int number = 0;

do{

Sout (‘give a number’);

Number = sc.nextInt();

Sout (‘here is the number :’);

Sout (number);

}while(number >= 0);

Sout (‘the end’);

Infinite loop = while(true)

BREAK AND CONTINUE :

Use of break –

Int i= 0;

While(true){

Sout(i);

i++;

if(i > 5){

break;

}}

Use of continue –

Int i= 0;

While(true){

If(i == 3){

i=i+1;

continue; } // this continue is used to skip and go back to second loop

Sout(i);

i++;

if(i > 5){

break;

}}

EXCEPTIONS HANDLING :

Int [] marks = {1,2,3};

//if there is a possibility of exceptions, then we can put that risky line in the try block

Try{

Sout(marks[5]);

}catch(Exception exception) {

}

Sout(“the name is aman”);}

METHOD and FUNCTIONS :

Method:

Public static void printJava(){

Sout(“hello Java”);

}

printJava(); //calling the function in the main class

printJava();

printJava();

Public static void printName(String name){

Sout(name);

}

printName(); //calling

public static void printsum(int a, int b){

int sum = a+b;

Sout(sum);

}

printSum(1,6); }}

MINI PROJECT :

Generate a random number and let the user to guess the number. If the user guess upper number than, it will show upper and same for the lower one. Then if the user gets tired, then just have to press -1 and let him know the result.

Public class Main{

Public static void main(String[] args){

int m

}}