TOKENS IN JAVA

TOKEN:

- TOKENS ARE SMALLEST ELEMENT OF OUR JAVA PROGRAM
- IN JAVA PROGRAMMING LANGUAGE EACH INDIVIDUAL WORD AND PUNCTUATION IS REFERRED AS TOKENS
- JAVA LANGUAGE SUPPORTS FIVE TOKENS

1. KEYWORD
2. IDENTIFIERS
3. CONSTANTS/LITERALS
4. SPECIAL SYMBOL
5. OPERATORS

- JAVA LANGUAGE IS COLLECTION OF TOKENS

Keywords:

1. Keywords are predefined words in Java. Keywords are those words whose meaning is already know to the Compiler
2. Various keywords in java are int,float, double, if, else, void, while, dowhile, for, abstract, final, enum, implement, extends, go to, break, continue, etc
3. They are 52 keywords in Java programming
4. The programmeres cannot use keywords as name of variable, methods, classes, or as any identifier

Identifier:

- **Identifier is name given to varibales, function, array, class etc**
- **Identifier are also called as variables**
- **Variable are memory boxes that holds values**
- **The variable name must begin with an alphabet or underscore followed by alphabets or numbers**
- **Eg: -_ test, test, sum2**

```
Int sum; //here sum is an identifier
Class Train
{
}//here train is an identifier
```

3) constant/literals

Constants in Java define fixed values, that does not change during the execution of a program

Eg:- final float pi=3.14;

Final int a=10;

Once the variable is declare as constant then we cannot change its value

4) Special symbol

Special symbol like [], (), {}, ;, *, #, $ etc are used in Java

Special symbols have special meaning that can not be used for some other purpose

5) Operators

An operators is a special symbol that perform an action when applied between Java variables

Eg:-+ , - , * , / , % etc

Sum =a +b;

A and b are operands , + is the additional operator, that performs the addition of the numbers. The result value will be stored in the variable SUM, = is the assignment operator

APPROACHES OF PROGRAMMING

They are two types Of approach

- **Procedural language**
- **Object oriented Language**

Procedural language:-

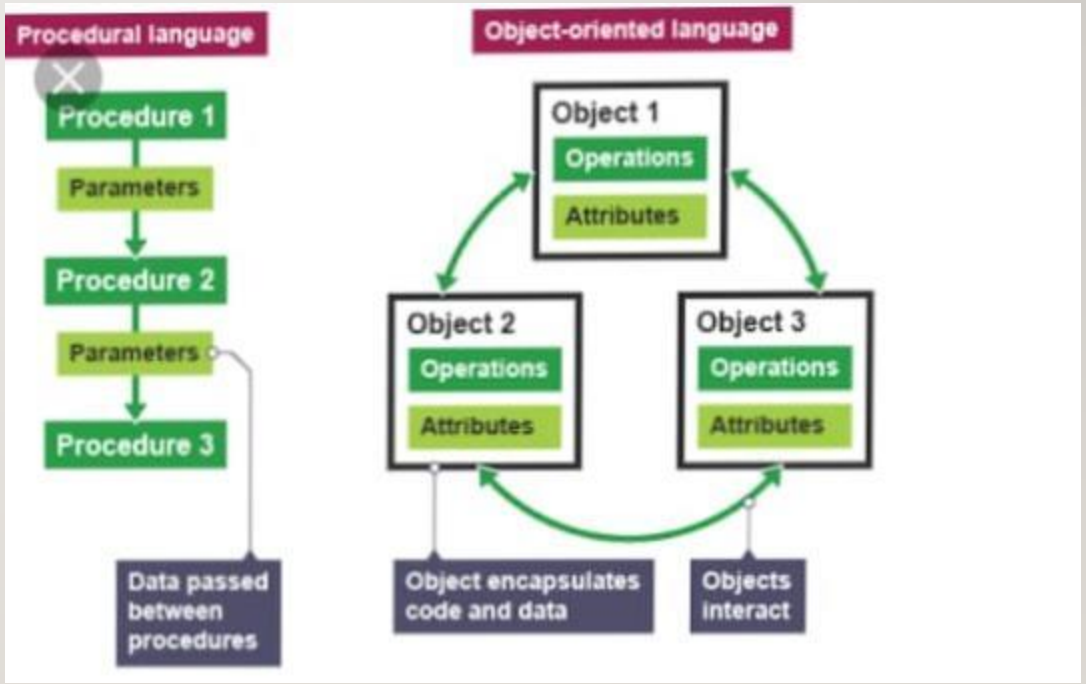
- **Top_down_approach**
- **Static_programming**
- **Focus_only_on_functionality**

Eg:- c program

Object oriented Language:-

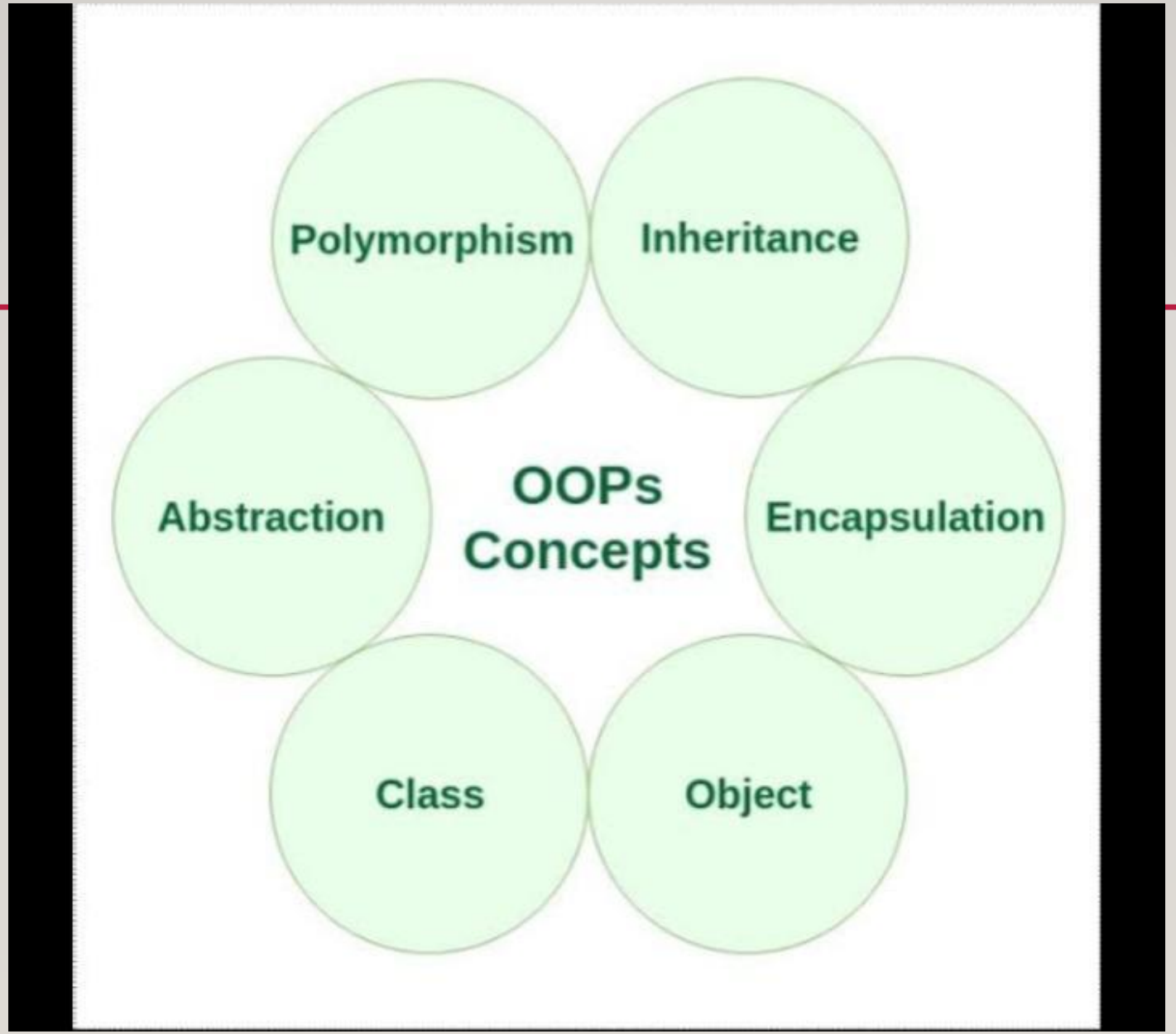
- Bottm up approach
- Dynamic programming
- Focus on object and classes

Eg:-Java, c#



OOPS CONCEPT

- Object oriented programming language manily designed to provide security and reuseability to the code
- Object oriented programming language is about creating object that contains both data and methods
- It provides a clear structure of program
- To call any programming language as object oriented programming language it must statisfies set of principles
- They are 6 main principles which statisfies object oriented programming language


Polymorphism
Inheritance
Abstraction
OOPs Concepts
Encapsulation
Class
Object

Abstraction:-

Hiding the internal implementation and highlighting the set of services that process is called abstraction

For eg :- bank ATM screen (hiding internal implementation and highlighting set of services like withdraw money , money transfer, balance etc)

Mobile phones person is hiding the internal circuit implementation and highlighting the touch screen and services

ENCAPSULATION:-

Wrapping of data members and members functions ain a special container called class is called encapsulation

Process of binding data member and member functions in a single unit is called encapsulation

Ex:-class Student

{ int roll no;

int age;//data member

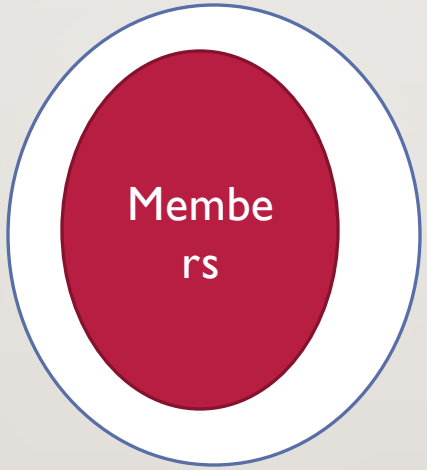
} +

Read() // member function

{}

Write()

{}



class

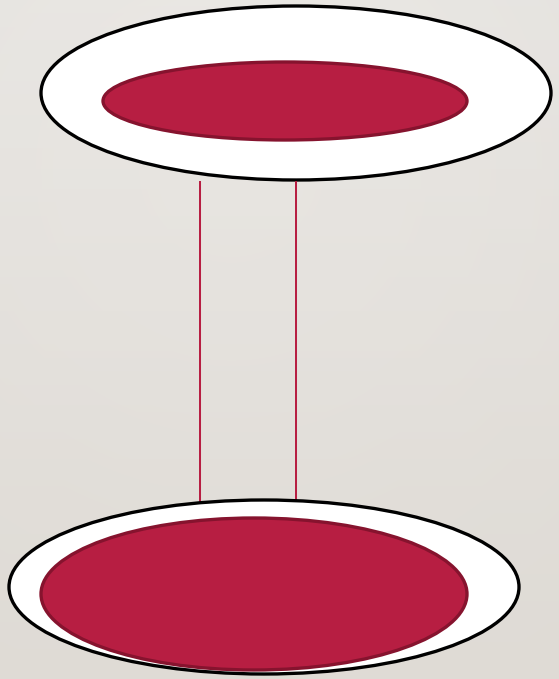
POLYMORPHISM

- Polymorphism is a "Greek word" The word polymorphism Means many forms
- Poly=many, morph=shape
- Changing the behavior of entities according to input that is supplied to them is known as polymorphism
- Ability to perform in different form is called polymorphism

INHERITANCE

- The process of getting properties and behavior from one calss to another class is called Inheritance

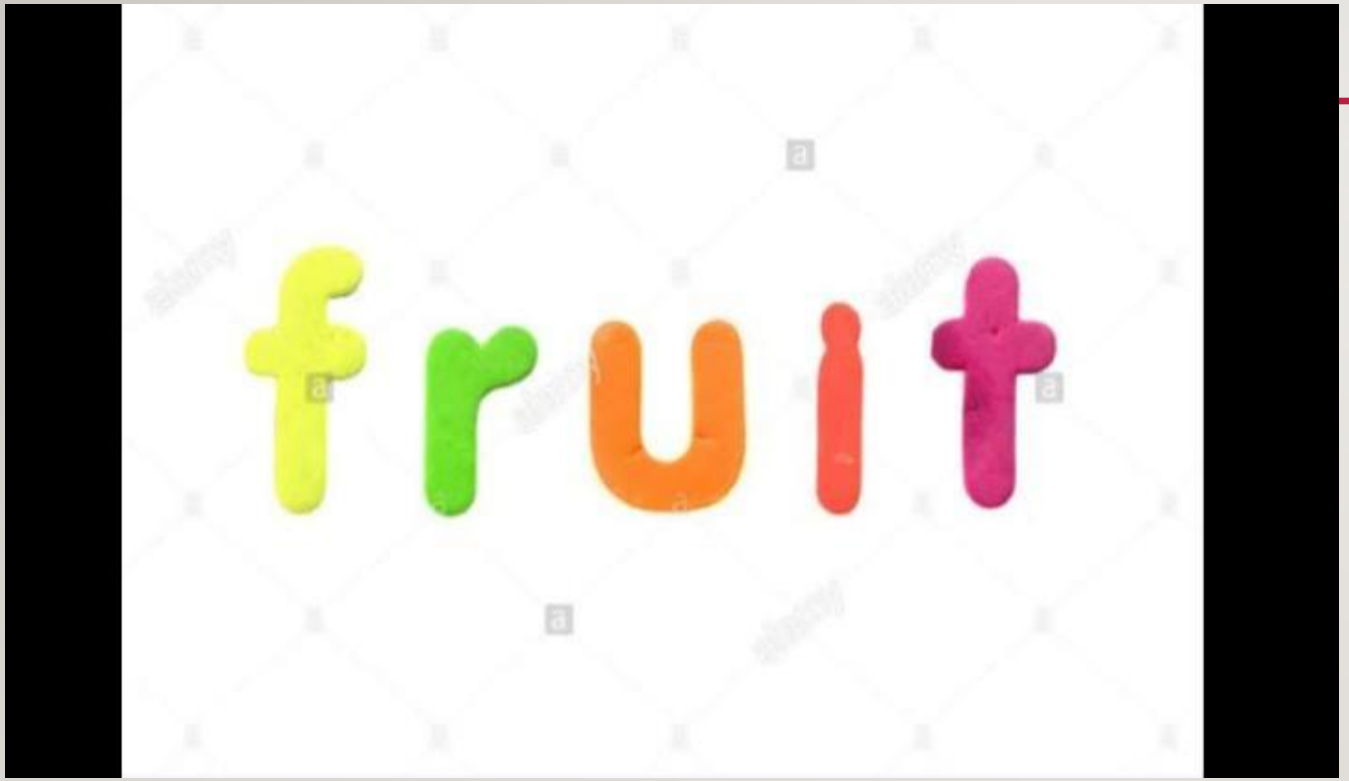
- Inheritance allows reuseability of code
- Properties : varibales
- Behaviour :methods
-

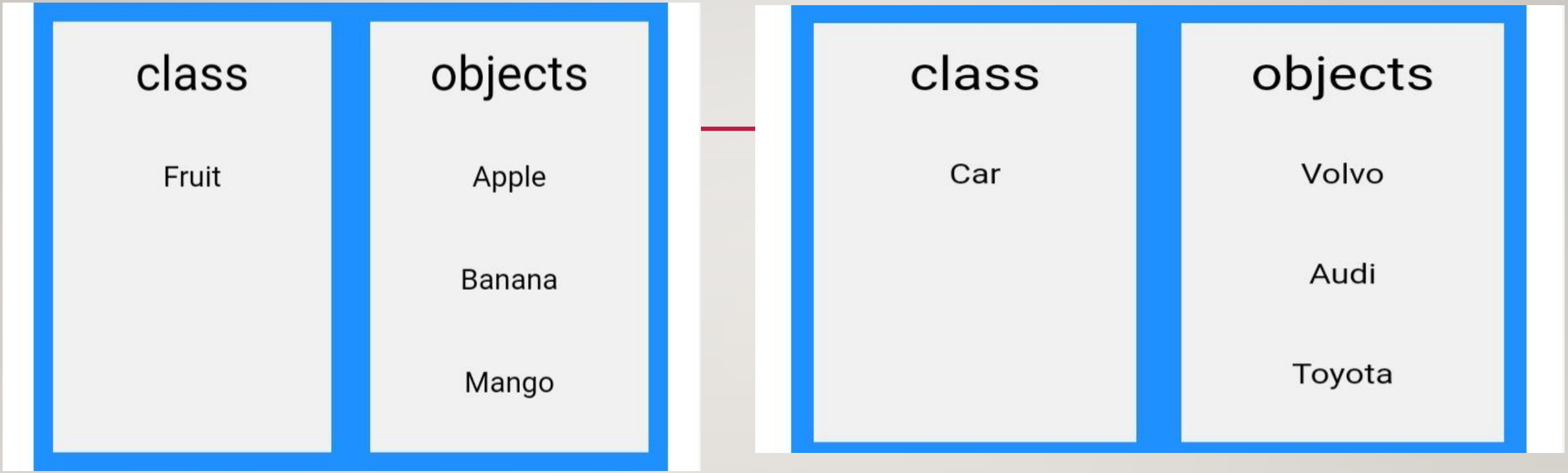


parent class/superclass/base class

Child class/subclass/derived class

Which is Object And which is class ?




class
Fruit
objects
Apple
Banana
Mango
class
Car
objects
Volvo
Audi
Toyota

CLASS:-

- A class is group of object that have common property

- A class is a template or blueprint from which types of object are created
- Class is a way of building a data &oriented method in a simple unit
- Class is a logical enity
- Class is a template of object

Eg:- class name Animal

Data member age, color, no of leg

Behavior eat(), walk(), sleep()

OBJECT:-

- Object is real world entity &physical entity

Object is nothing but instance of a class

- Object is a varibale of class
- Every object contains 3 characteristics

State (represent data of an object)

Behavior (represent behavior of an object)

Identify (used to identify the object uniquely)

Eg:- pen

Name : cello

Color: blue// state

Behaviour : use to write

STATEMENT IN JAVA :-

- Statement are everthing that make up a complete unit at execution
- Int x= 10*3;

here 10*3 is an expression that return 30 and int x =10*3 is statement

EXPRESSION IN JAVA:-

- Expression are constructed from operands and operators
- An expression is a statement that convey some value

- Eg:- sum a+b;

Types of expression:-

Based on operator and operand used an expression can be classified into several 6 types

1. Integer expression
2. Real expression
3. Arithmetic expression
4. Relational expression
5. Assignment expression
6. Logical expression

1) Integer expression:-

An expression which contain integer and operator

Eg :-A=3*5;

2) Real expression:-

An expression which contains floating point value and operator

A=3.14*a*a;

3) Arithmetic expression:-

An expression which contain operands and arithmetic operator is called arithmetic expression

Sum=a+b;

4) Realational expression:-

An expression which contain relational operator and operands

Eg:- x=if (a>b&&a>c)

5) Assignment expression:-

An expression which contain assignment operator and operand

Eg:- x=10;

Null=a*b;

6) Logical expression:-

An expression which contain logical operator and operand

Eg:- a=! B;

OPERATORS IN JAVA :-

An operator is symbol that is used to operate on value Or variable

An operator trlls thr compiler to perform certain specific operation

TYPES OF OPERATOR IN JAVA:-

1) Arithmetic operator
2) Assignment operator
3) Relational operator
4) Logical operator
5) Bitwise Operator
6) Conditional operator
7) Increment or Decrement

ARITHMETIC OPERATOR:-

The arithmetic operator perform mathematical calulation like addition, subtraction, multiplication, division and modula

Operator	Example	Output
+	2+3	5
-	5-2	3
*	3*2	6
/	6/2	3
%	6%2	0

ASSIGNMENT OPERATOR:-

This operator assigns value to a varibale

Operator	Example	Meaning
= (equal to)	**A=5;**	**The values assignment to the varibale a**
+= (short hand assignment operator)	X+=y;	X=x+y;
-=	**X-=y;**	X=x-y;

RELATIONAL OPERATOR:-

- The relational operator is used to relate two varibales

Operator	Example
<	A<b
>	A>b
==	A==b
! =	A! =b
<=	A<=b
>=	A>=b

A=5, b=6;
(A<b) =true
(A>b) =flase
(A==b) =flase

LOGICAL OPERATOR:-

- Logical operator is used to compare two or more expression

Operator	Example
&&	(A>b) &&(a>c)
||	(B>a) ||(b<c)
!	! (A>c)

Operator	Meaning	Example	Result
&&	Logical and	(5<2)&&(5>3)	False
||	Logical or	(5<2)||(5>3)	True
!	Logical not	!(5<2)	True

CONDITIONAL OPERATOR:-

- It is also Called ternary operator this operator is used to check conditions
- This operator takes 3 arguments

Operator ? :

Syntax :- exp 1? Exp2: exp 3;

If exp 1 is true then exp2 executed

If exp 1 is false then exp 3executed

a=1, b=10

x=(a<b)? a*a: b%a;

=(1<10) ? 10 : 0

=10

BITWISE OPERATOR:-

- Bitwise Operator works on bits and perform bit by bit operations

P	Q	P&q	P|q	P^q
0	0	0	0	0
0	I	0	I	I
I	0	0	I	I
I	I	I	I	0

INCREMENT /DECREMENT:-

- It is used to increment or decrement the value by 1

- They are called uniary operator because it is applied on only one operand
- Example :- a= 5;

++a,	a++,	--a,	a--
6	5	4	5

CONSTANT IN JAVA :-

- Constant is a varibale whose value cannot be changed once

It has been assigned

- Java does not have built in support for constant
- To define a varibale as constant, we just need to add a keyword
- Final in front of the variable declaration
- Eg:- final float pi=3.14;

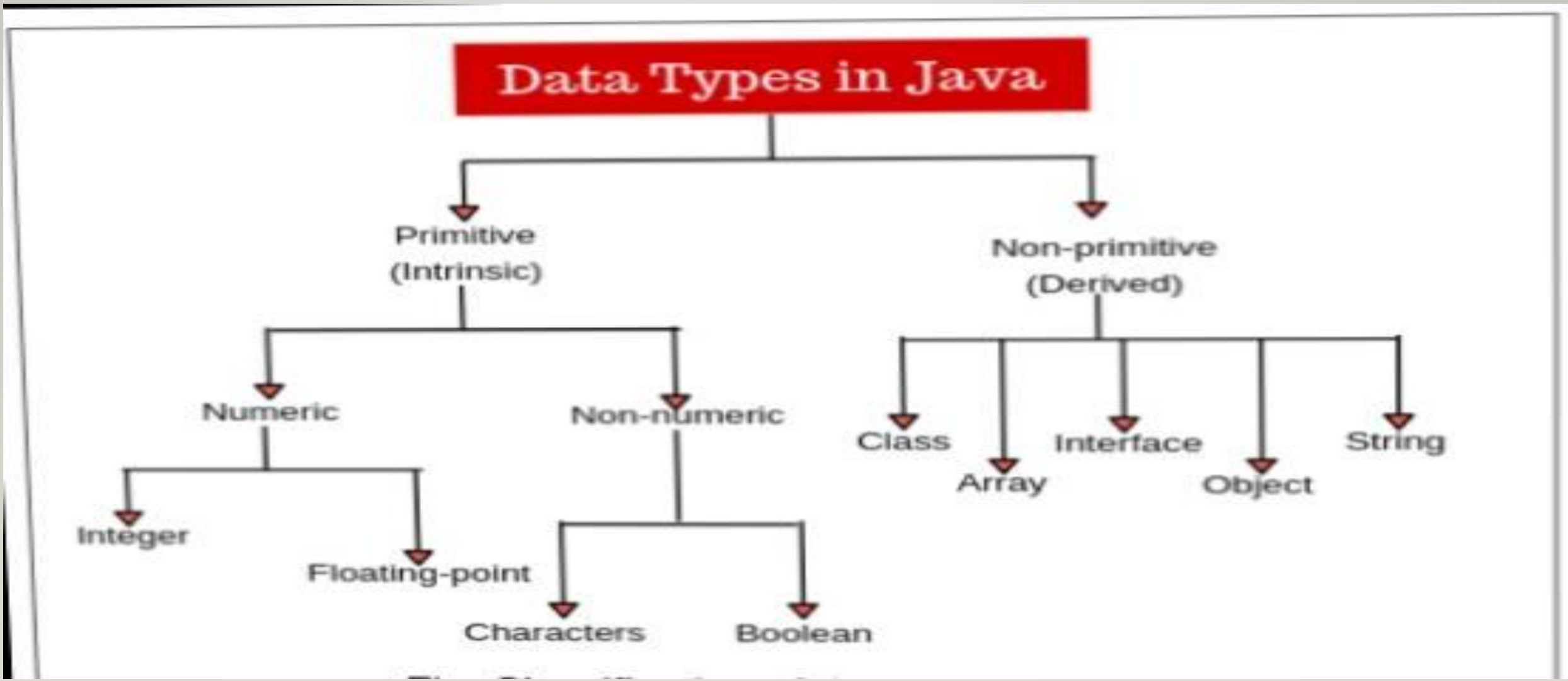
  Final int a=10;

DATA TYPE :-

- Data type specifies the different sizes and values that can be stored in a variable there are two types of data types in java

1. Primitive datatype (or) predefined data type
2. Non primitive datatype

DATA TYPES IN JAVA



RANGES OF DATA TYPE:-

TYPE	DESCRIPTION	DEFAULT	SIZE	EXAMPLE LITERALS	RANGE OF VALUES
boolean	true or false	false	1 bit	true, false	true, false
byte	twos complement integer	0	8 bits	(none)	-128 to 127
char	unicode character	\u0000	16 bits	'a', '\u0041', '\101', '\\', '\'', '\n', 'β'	character representation of ASCII values 0 to 255
short	twos complement integer	0	16 bits	(none)	-32,768 to 32,767
int	twos complement integer	0	32 bits	-2, -1, 0, 1, 2	-2,147,483,648 to 2,147,483,647
long	twos complement integer	0	64 bits	-2L, -1L, 0L, 1L, 2L	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	IEEE 754 floating point	0.0	32 bits	1.23e100f, -1.23e-100f, .3f, 3.14F	upto 7 decimal digits
double	IEEE 754 floating point	0.0	64 bits	1.23456e300d, -1.23456e-300d, 1e1d	upto 16 decimal digits

STRUCTURE OF JAVA PROGRAM:-

- Every programing language to write a program we need to follow standard format for development of application
- Sun microsystem developers as describe following structure for writing java application

Package details;

Class<class Name)

{Data members

Member function

Public static void main (string args[])

{Block of statement;

}}

System. out. Println()

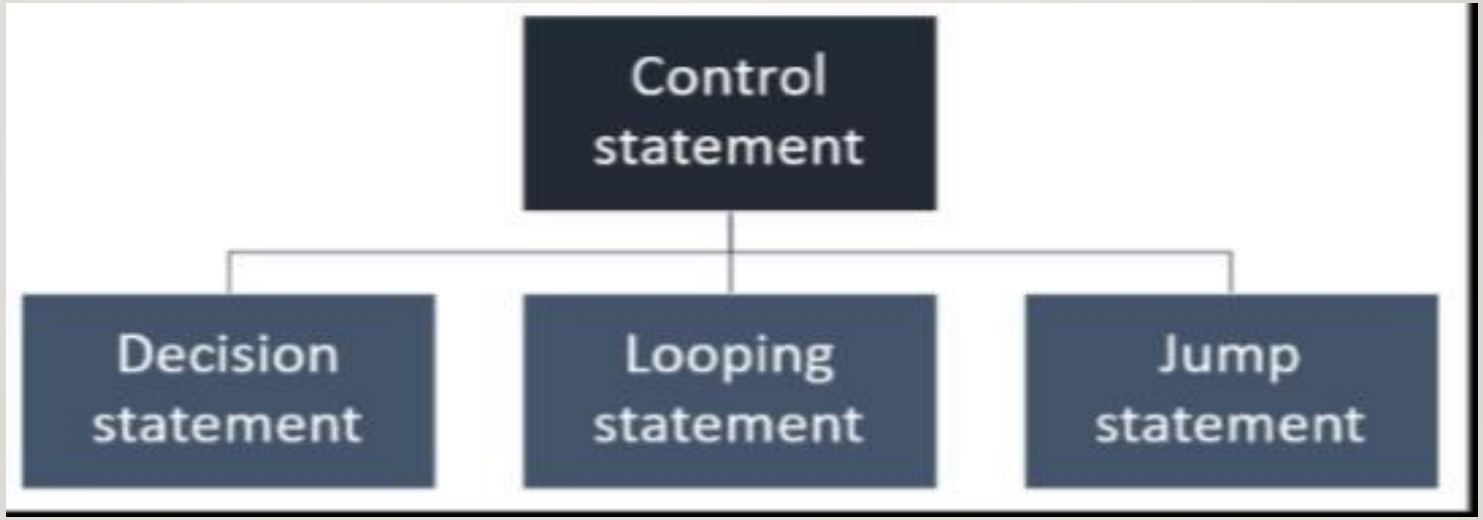
System is a class in java. lang package

Out is a static variable (object) present in a system class of type printstream

Println() is a method presents in printstream

CONTROL STATEMENTS IN JAVA :-

- A flow control in java describes order in which the statement will executed at runtime
- A flow control statement injava can be classified into three types



DECISION CONTROL:-

- Decision control have one or more conditions to be evaluated or tested by the program, along with statement or statements

1. If statement
2. If else
3. If else –ladder
4. Nested if else
5. Switch

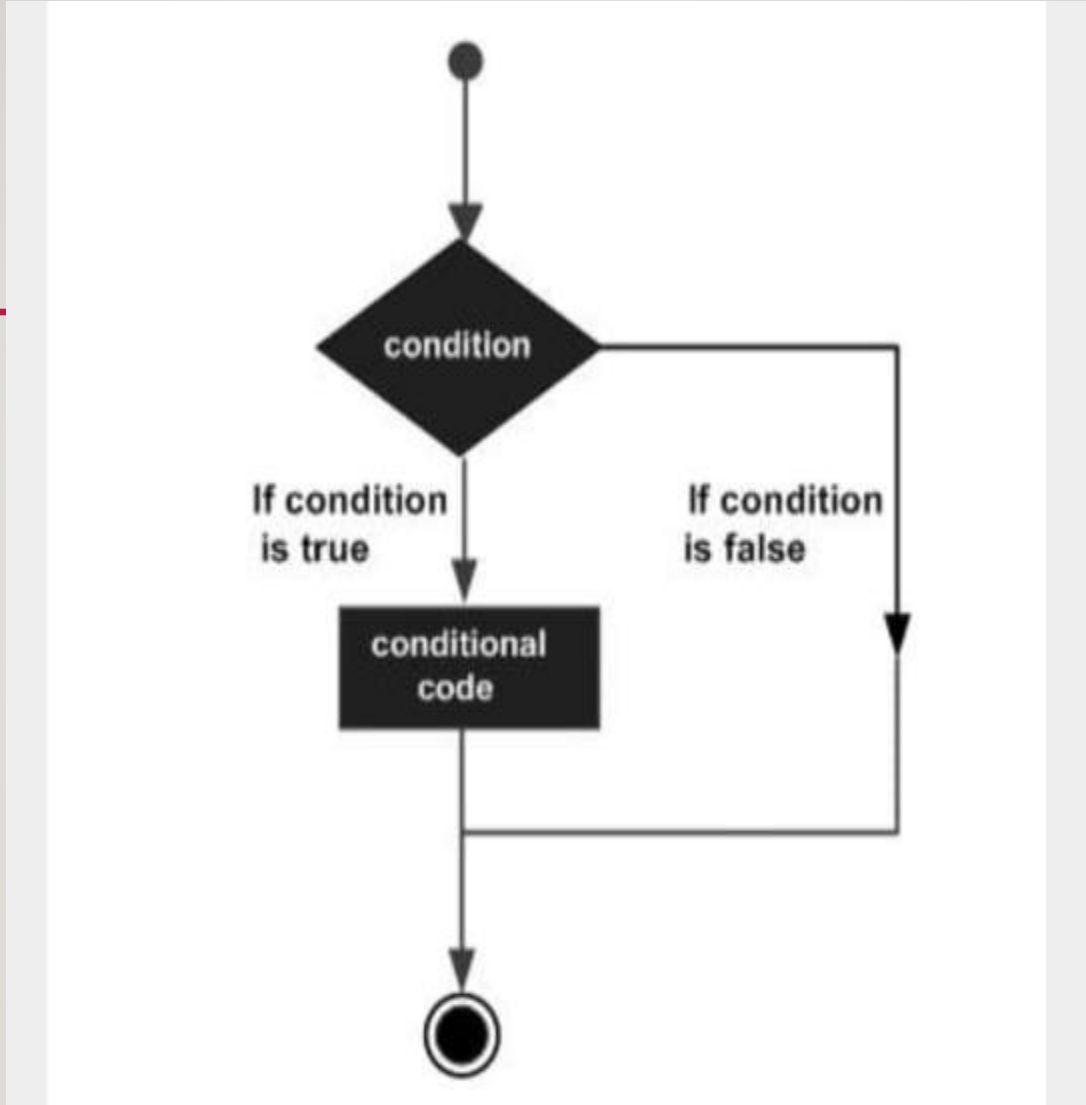
IF STATEMENT:-

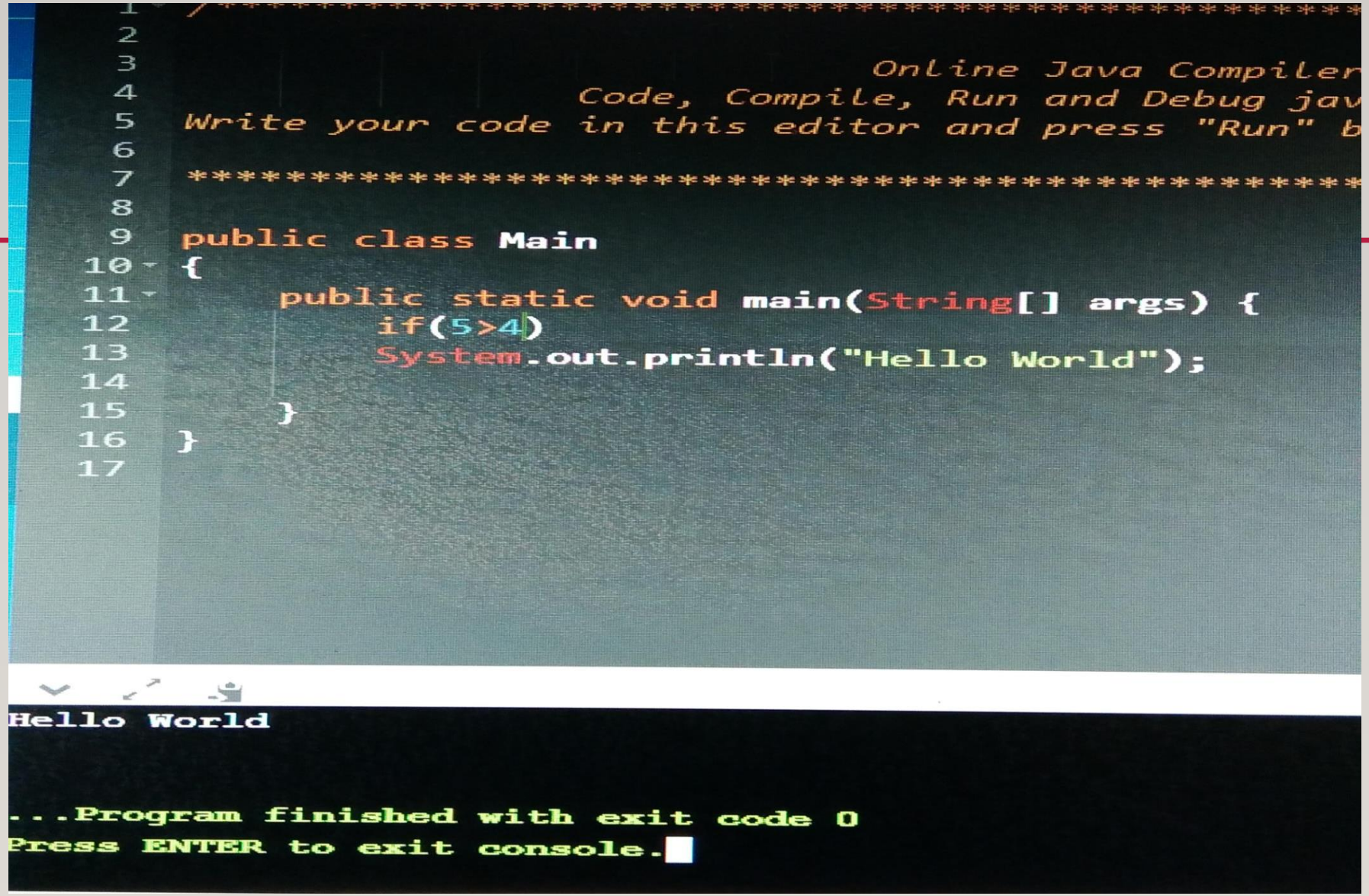
- The keyword if is used to execute a statement or block of statements if condition is true

Syntax;-

If (condition)
{Statement;}

if(condition)
{ statement 1;
Statement 2;}


condition
If condition
is true
If condition
is false
conditional
code


/*****
Online Java Compiler
Code, Compile, Run and Debug jav
Write your code in this editor and press "Run" b
*****
public class Main
{
public static void main(String[] args) {
if(5>4)
System.out.println("Hello World");
}
}
Hello World
...Program finished with exit code 0
Press ENTER to exit console.

IF ELSE STATEMENT:-

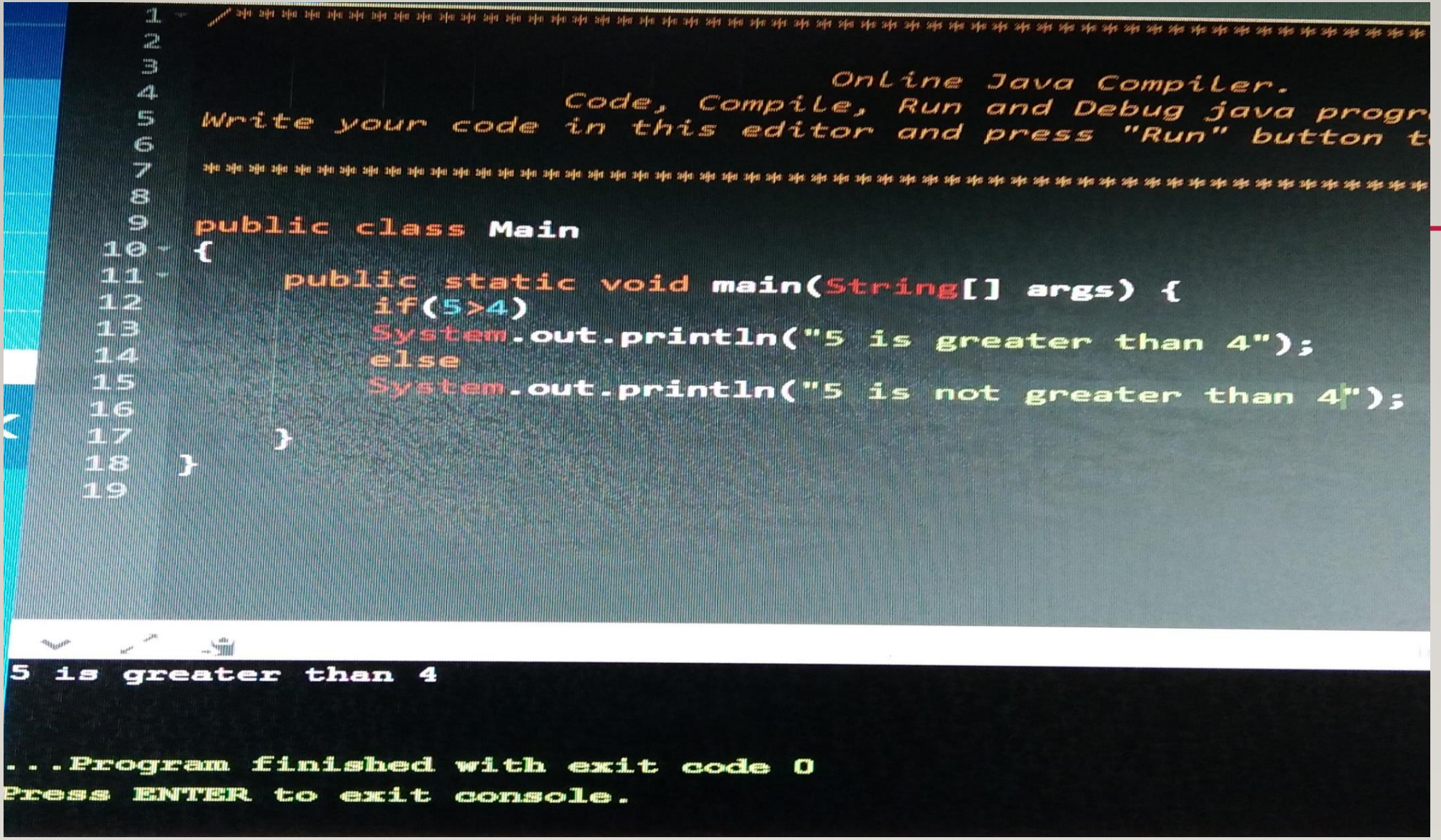
- In this decision control statement we have two blocks of statement

---

- If condition results true then if block will be executed else the statement inside else block will be executed
- Else block cannot exist without if condition

syntax:-

```
if(condition)
{ statement;}
Else
{ statement;}
```


/*
Online Java Compiler.
Code, Compile, Run and Debug java progr
Write your code in this editor and press "Run" button t
*/
public class Main
{
public static void main(String[] args) {
if(5>4)
System.out.println("5 is greater than 4");
else
System.out.println("5 is not greater than 4");
}
}
5 is greater than 4
...Program finished with exit code 0
Press ENTER to exit console.

SWITCH CASE STATEMENT:-

- Switch is a decision control statement that allows us to select one among multiple decision
- Switch test the value against list of integer or character values each value are called case
- When match is found then statement associated with the value is executed

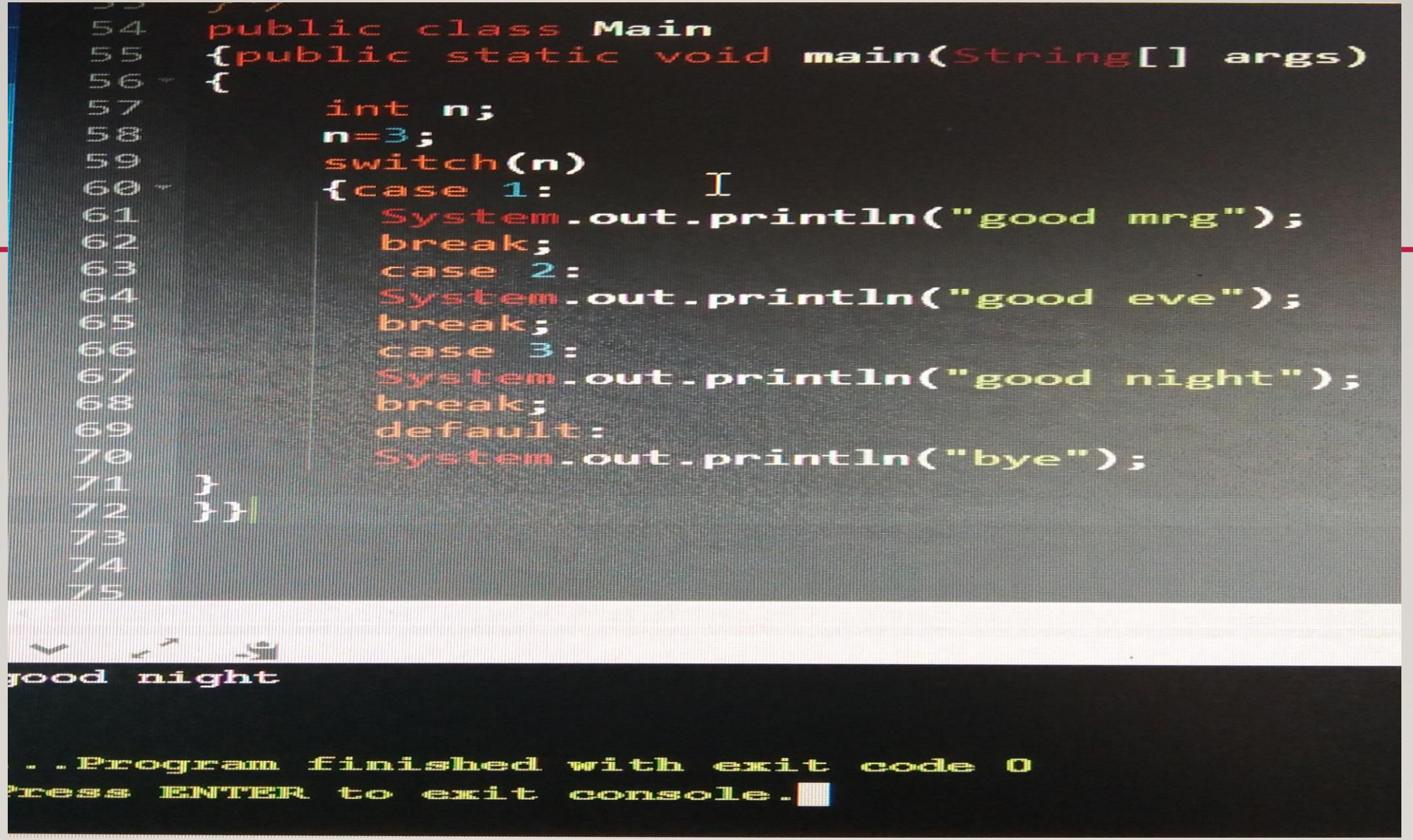
Rules of switch case :-

- Expression provided in switch keyword should be an integer variable

- Case level must be unique
- Case level must end with collon :
- Must be in integer type
- Must be in integer type
- Must not be floating value
- Default label is optional
- If no match is found than only statement after default is executed
- Break statement is used to execute from the switch

Syntax:-

```
Switch (condition)
{   case1 :
      Statement;
      break;
       Case2:
       Statement;
        break;
       Default:
       Statement;
}
```

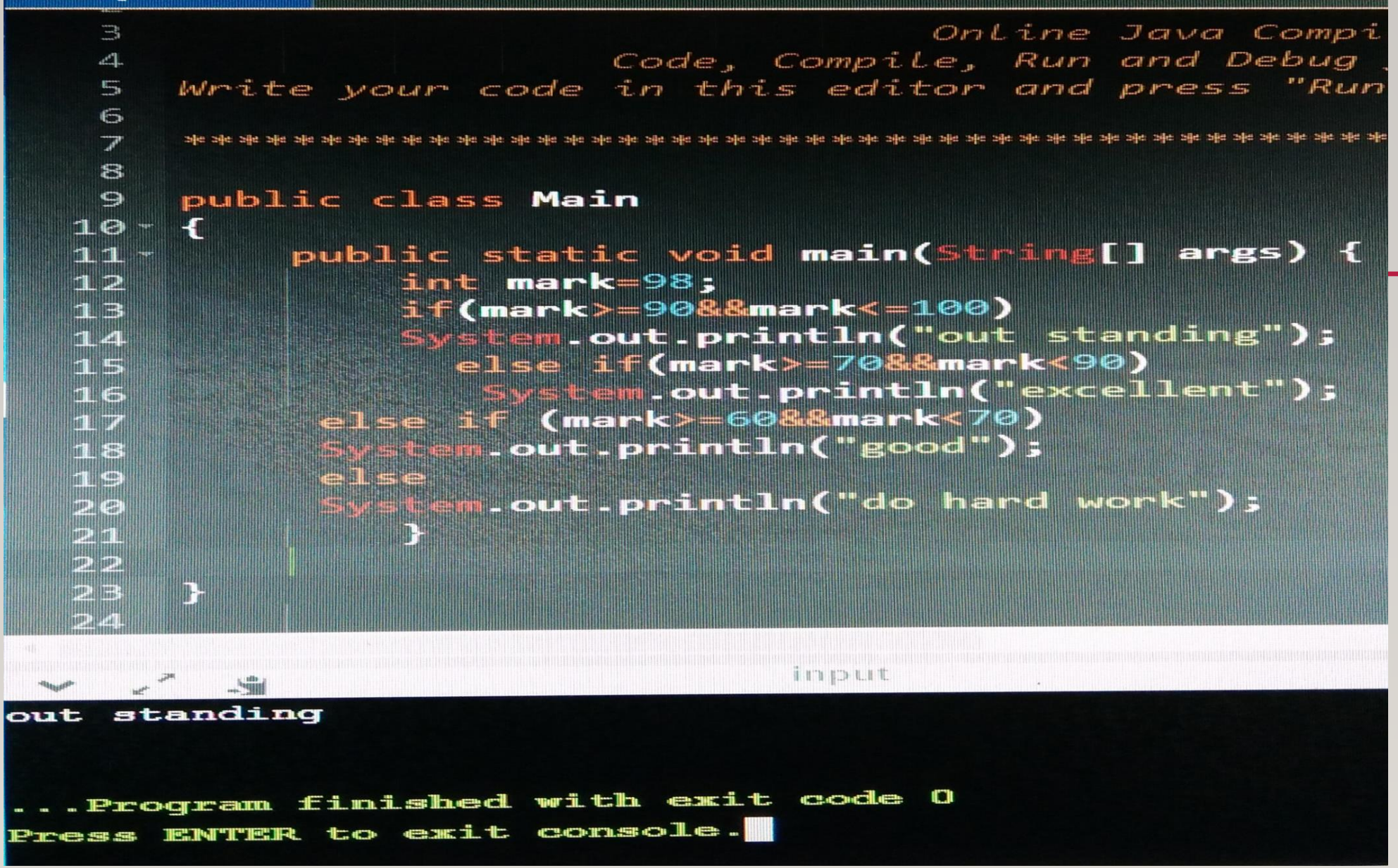

54 public class Main
55 {public static void main(String[] args)
56 {
57 int n;
58 n=3;
59 switch(n)
60 {case 1:
61 System.out.println("good mrg");
62 break;
63 case 2:
64 System.out.println("good eve");
65 break;
66 case 3:
67 System.out.println("good night");
68 break;
69 default:
70 System.out.println("bye");
71 }
72 }}
good night
. . Program finished with exit code 0
Press ENTER to exit console.

IF ELSE –LADDER:-

- In this decision control statement we have more than two blocks of statement
- In java if we have multiple condition to execute then we use if else ladder

Syntax:-

```
If (condition)
Statement ;
Else if (condition)
Statement;
Else if (condition)
Statement;
Else
Statement;
```

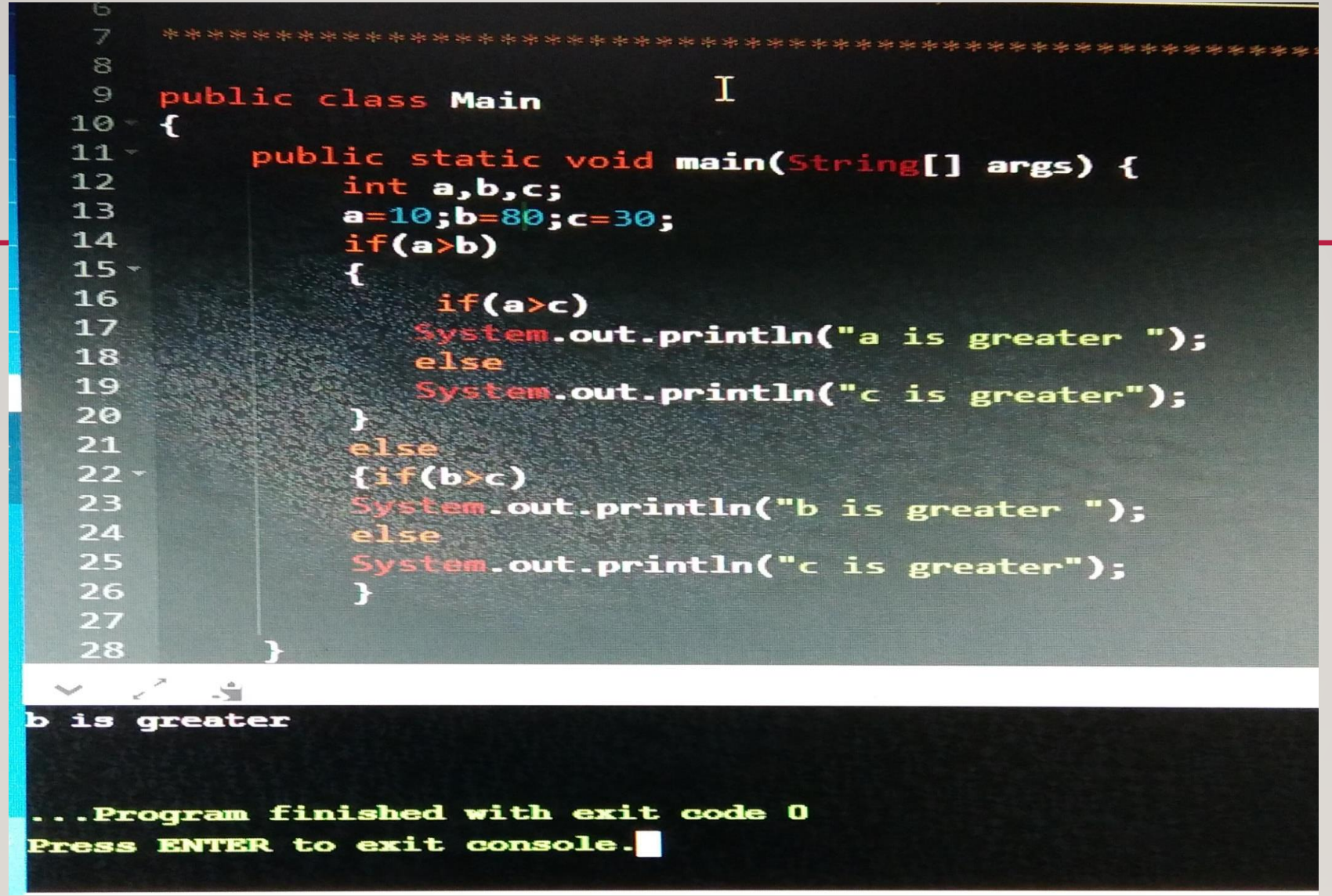

Online Java Compi
Code, Compile, Run and Debug
Write your code in this editor and press "Run
public class Main
{
public static void main(String[] args) {
int mark=98;
if(mark>=90&&mark<=100)
System.out.println("out standing");
else if(mark>=70&&mark<90)
System.out.println("excellent");
else if (mark>=60&&mark<70)
System.out.println("good");
else
System.out.println("do hard work");
}
}
input
out standing
...Program finished with exit code 0
Press ENTER to exit console.

NESTED IF ELSE:-

- A nested if is an if statement that is the target of another if or else.
- Nested if statement means an if statement inside an if statement
- Yes java allows us to nest if statement with I'm if statement
- If none of the conditions is true, then the final else statement will be executed

SYNTAX:-

```
If (condition)
{
If(condition)
Statement;
Else
Statement ;
}
Else
{ if (condition)
Statement;
Else
Statement;
}
```


public class Main
{
public static void main(String[] args) {
int a,b,c;
a=10;b=80;c=30;
if(a>b)
{
if(a>c)
System.out.println("a is greater ");
else
System.out.println("c is greater");
}
else
{if(b>c)
System.out.println("b is greater ");
else
System.out.println("c is greater");
}
}
b is greater
...Program finished with exit code 0
Press ENTER to exit console.