# Creating a
# JAVA
# file

create it with .java extenstion

## Boilerplate code:

```
public class filename {
        public static void main (String args[]) {
          code
        }
}
```

## Output in Java:

```
System.out.print ("Output");
```

## How to compile:

compile → javac filename.java

Run → java filename.java

# Output in Java

```
system.out.print ("   ");

system.out.println("   ");
```

↳ gives next line in output

# Variables in Java

Declaring:

```
int      a ;
```

data type   identifiers

# Data types in Java

Primitive → already exist

| | | |
|---|---|---|
| byte | 8 bits | (-128 to 127) |
| short | 2 bytes | |
| char | 2 | |
| boolean | 1 | |
| int | 4 | ±1 billion |
| long | 8 | |
| float | 4 | |
| double | 8 | |

Non-primitive → we have to create them

- String
- Class
- Array
- Object
- Interface

# Type Conversion

or widening conversion
or implicit conversion

Conversion happens when:

(a) type compatible

byte → short → int → float → long → double

(b) size of destination type > size of source type.

It happens automatically.

# Type Casting

or narrowing conversion
or explicit conversion

eg:

```
float a = 25.9999f;
int b = (int) a;
          ↳ Type casting
System.out.println(b);
```

output: 25

# Type promotion in expressions

1) Java automatically converts each

byte, short or char → <u>int</u>

2) Per agar koi bhi ek operand long

long, float ya double hai

then whole expression is converted into

long float & double.

# Input in Java

next
nextLine
" Int
" Float
" Byte
" Double
" Boolean
" Short
" Long

Write this on top ⤵

```
import java.util.*;

public class JavaBasics {
             file name

    public static void main(String args []){

    Scanner     sc    =new Scanner
     class    object        (System.in);


    int a= sc.nextInt ();

    }
}
```

# Operators in Java

$\underbrace{a}_{} + \underbrace{b}_{}$ operands

## Arithmetic

+

−

*

/

%

## Unary

++

--

| Pre increament | Post increament |
|---|---|
| ++a /--a | a++ /a-- |
| value change then use | first use then change. |

## Relational

| | |
|---|---|
| == | equal to |
| != | not equal to |
| > | |
| < | |
| >= | |
| <= | |

## Logical operators

| | | | |
|---|---|---|---|
| && | logical AND | | |
| \|\| | logical OR | | |
| ! | Logical NOT | → | true → false<br>false → true |

## Assignment

=

+=

-=

*=

/=

# Conditional Statements

if
else
else if } same as before

## Ternary operators

```
variable = condition ? statement 1 : statement 2;
                              ↓              ↓
                             true          false
```

## Switch Statement

Same as before

```
switch (n) {
      case 1 :          ;
              break ;
      case 2 :          ;
              break ;
      default :         ;
}
```

# Loops

while

for

do while

all same as befor

for square root

Math.sqrt(n)