Java CheatSheet

For Beginners

HelloWorld.java ← File Name

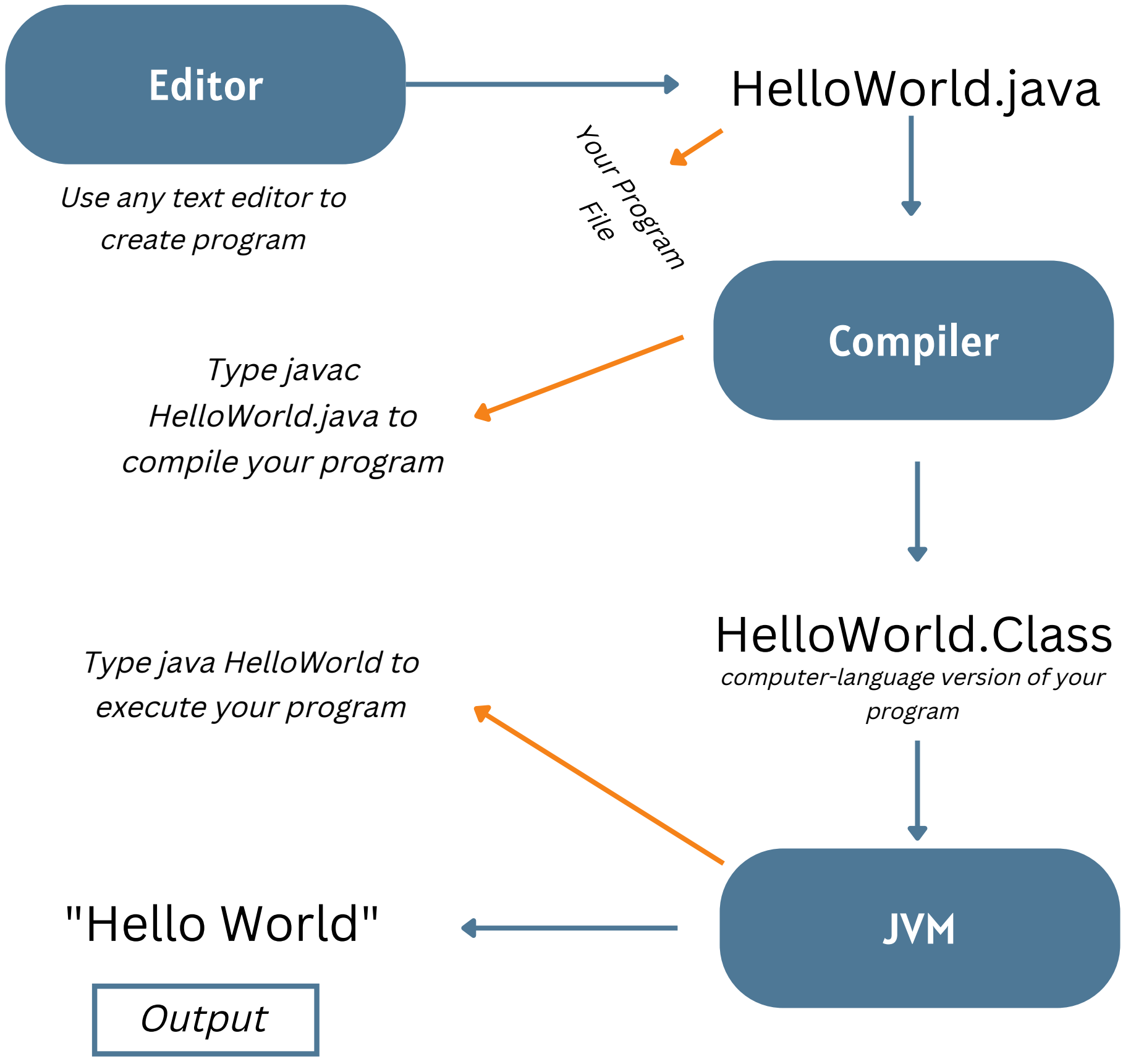
Class Name

Main Method

```
public class HelloWorld {

    public static void main(String[] args) {

        //Print "Hello World " in the terminal window
        System.out.println("Hello World")

    }
}
```

Statement

Editing, compiling, and executing.

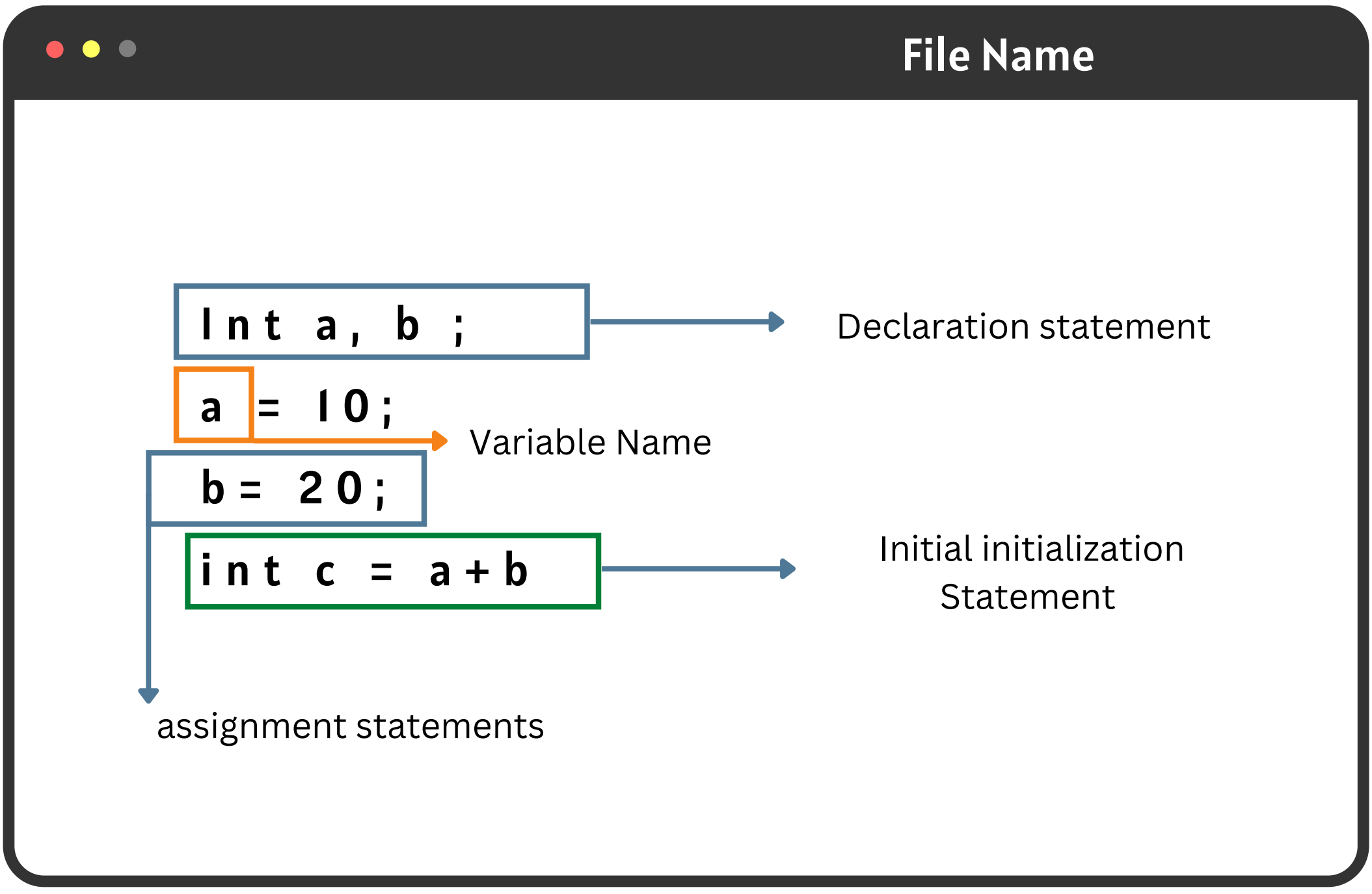


Data Types and Examples

Type	*Set of Values*	*Common Operators*
Int	Integers	+ - * / %
double	floating point numbers	+ - * /
boolean	boolean value	&& || !
char	characters	
String	sequence of characters	+

Int	:	74, 99, 21478521
double	:	3.5, 2.55, 6.0222e8
boolean	:	true , flase
char	:	'A' , 'D , 'F ' , '1' , '%' , '/n'
String	:	"AB" , "Hello"

Declaration and assignment statements



Integers

Values	Integers between -2^31 and +2^31-1
Examples	1234, 24, 5, 200000
Operations	sign add subtract multiply divide remainder
operators	+- + - * / %

Floating-point numbers.

Values	Real Numbers (Specified by IEEE 754 Standard))
Examples	3.14159, 2.0, 1.4142556568, 6.022e23
Operations	add subtract multiply divide
operators	+ - * /

Floating-point numbers.

Values	true or flase
literals	True / Flase
Operations	and or not
operators	&& || !

Comparison operators.

op	*Meaning*	*True*	False
==	equal	2 == 2	2 == 5
!=	not equal	3 != 2	2 ! = 2
<	less then	2 < 13	2 < 1
<=	less then or equal	3<=4	3 <= 2
>	greater thenl	14 > 5	5 > 14
>=	greater then or equal	3 > = 2	2 >= 3

Printing

void System.out.print(String s) print s

void System.out.println(String s) print s
Followed by new line

void System.out.println() print a new line

if and if else statement

absolute value	`if (x < 0) x = -x;`

put the smaller value in x and the larger value in y	`if (x > y)`
	`{`
	`  int t = x;`
	`  x = y;`
	`  y = t;`
	`}`

maximum of x and y	`if (x > y) max = x;` `else  max = y;`
error check for division operation	`if (den == 0) System.out.println("Division by zero");` `else  System.out.println("Quotient = " + num/den);`

error check for quadratic formula	`double discriminant = b*b - 4.0*c;`
	`if (discriminant < 0.0)`
	`{`
	`  System.out.println("No real roots");`
	`}`
	`else`
	`{`
	`  System.out.println((-b + Math.sqrt(discriminant))/2.0);`
	`  System.out.println((-b - Math.sqrt(discriminant))/2.0);`
	`}`

Nested if-else statement.

```
// Outer if statement.
  if(condition)
  {
// Inner if statement defined in outer if else statement.
      if(condition)
            statement1;
  }
// Else part of outer if statement.
  else {
        statement2;
    }
For example:
if (x > y)
{
  if (y > z)
    System.out.println("x is greater than y and z"); //
statement1.
}
else
  System.out.println("x is less than or equal to y"); //
statement2.
```

if-else if Ladder Statements in Java

```
if(condition)
   statement1;
else if(condition)
   statement2;
else if(condition)
   statement3;
   ...
else
   statement4;
```

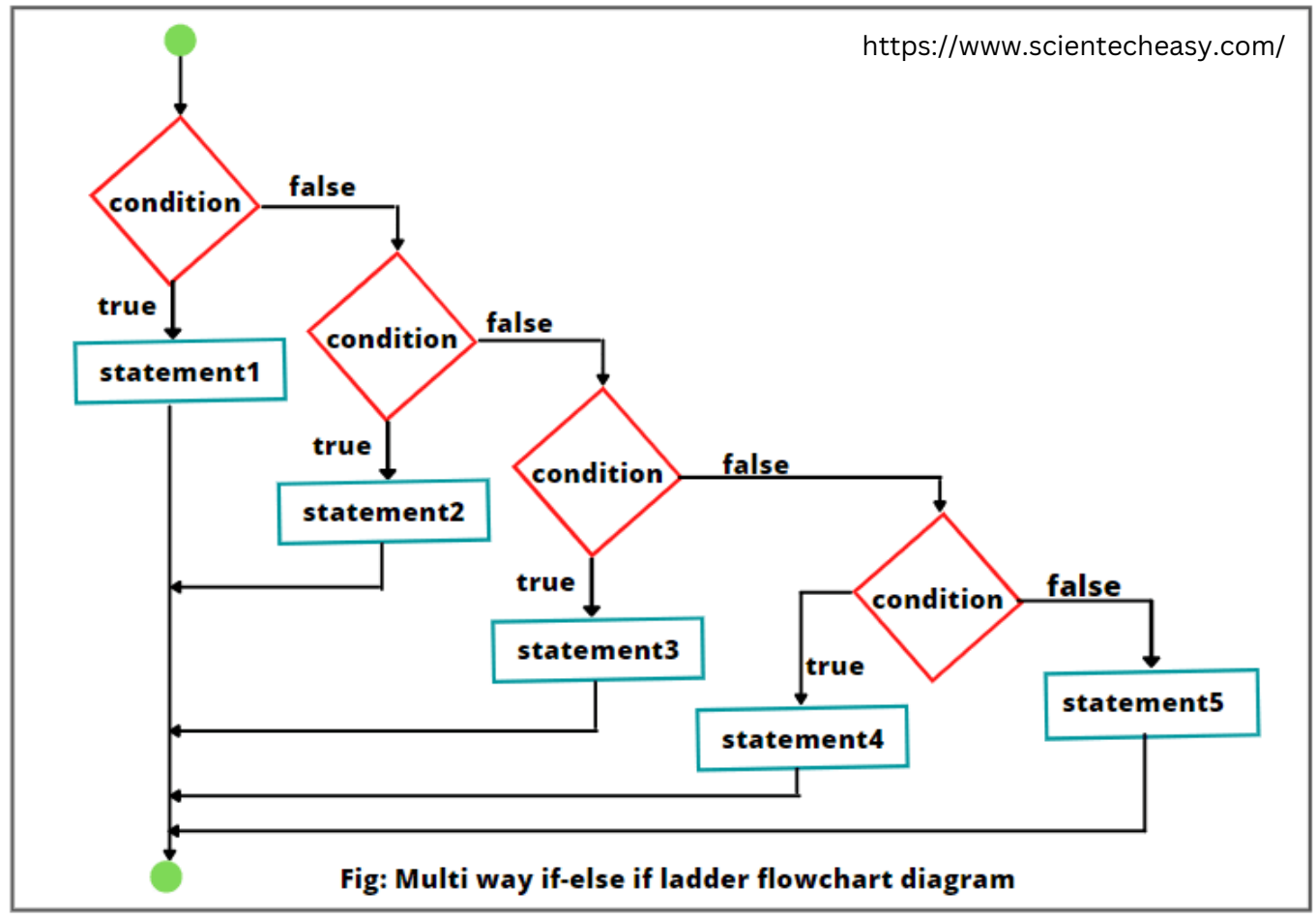


Fig: Multi way if-else if ladder flowchart diagram

while loop.

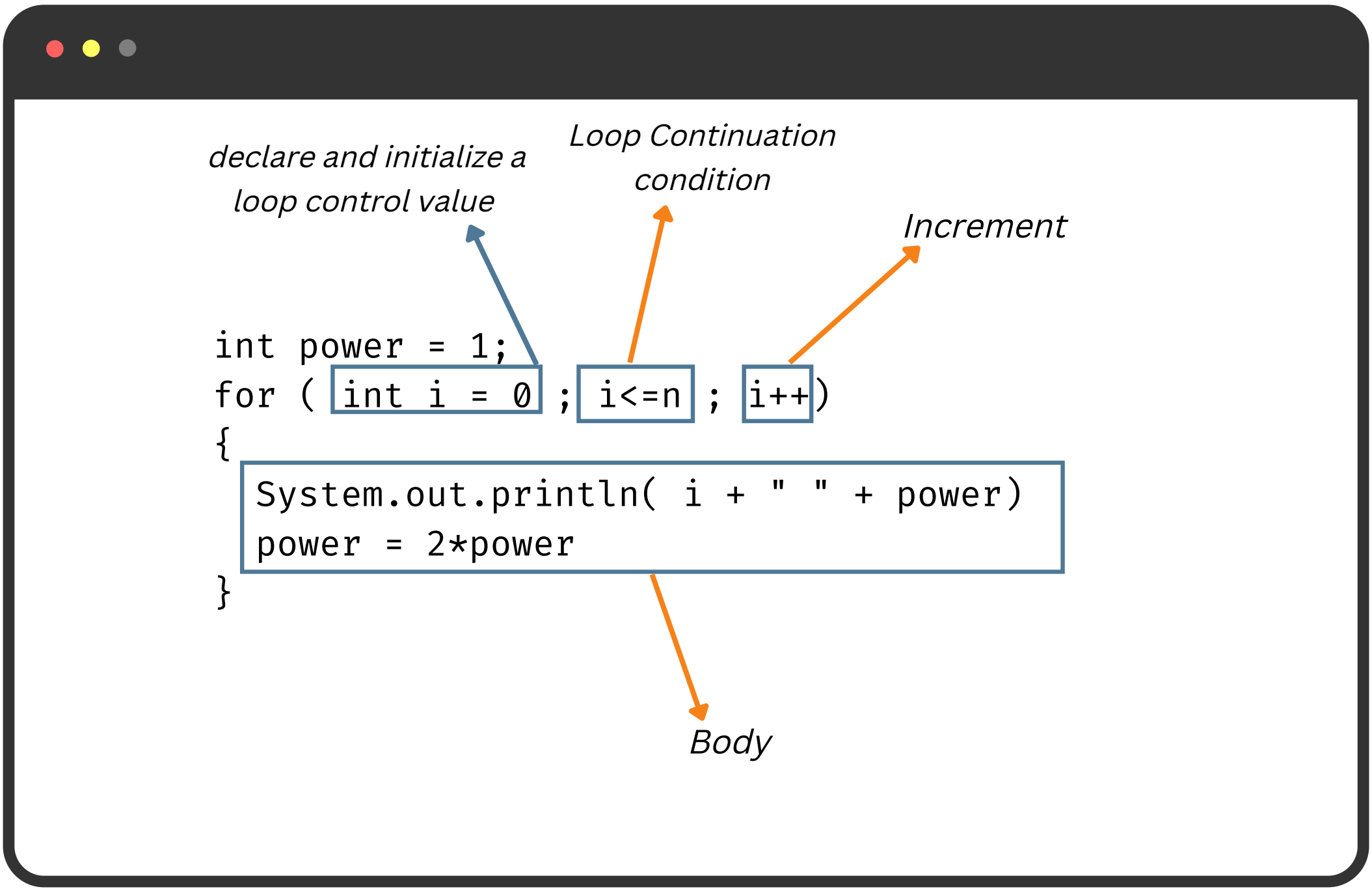
Loop Continuation condition

```
int power = 1;
while ( power <= n/2)
{
    power = 2*power;
}
```

Brackets are option when body is a single statement

BODY

For Loop


declare and initialize a loop control value
Loop Continuation condition
Increment
int power = 1;
for (int i = 0 ; i<=n ; i++)
{
System.out.println(i + " " + power)
power = 2*power
}
Body

Loops.

compute the largest power of 2 less than or equal to n

```
int power = 1;
while (power <= n/2)
   power = 2*power;
System.out.println(power);
```

compute a finite sum $(1 + 2 + \ldots + n)$

```
int sum = 0;
for (int i = 1; i <= n; i++)
   sum += i;
System.out.println(sum);
```

compute a finite product $(n! = 1 \times 2 \times \ldots \times n)$

```
int product = 1;
for (int i = 1; i <= n; i++)
   product *= i;
System.out.println(product);
```

print a table of function values

```
for (int i = 0; i <= n; i++)
   System.out.println(i + " " + 2*Math.PI*i/n);
```

compute the ruler function (see PROGRAM 1.2.1)

```
String ruler = "1";
for (int i = 2; i <= n; i++)
   ruler = ruler + " " + i + " " + ruler;
System.out.println(ruler);
```

Switch statement.

```
switch(expression){
case value1:
 //code to be executed;
 break;  //optional
case value2:
 //code to be executed;
 break;  //optional
......

default:
  code to be executed if all cases are not
matched;
}
```

Arrays

Single Dimensional Array in Java

Syntax to Declare an Array in Java

```
dataType[] arr; (or)
dataType []arr; (or)
dataType arr[];
```

Instantiation of an Array in Java

```
arrayRefVar=new datatype[size];
```

Arrays

Single Dimensional Array in Java

```
int a[]=new int[5];//declaration and
instantiation

a[0]=27;//initialization
a[1]=25;
a[2]=75;
a[3]=47;
a[4]=59;
```

Arrays

Multidimensional Array in Java

Syntax to Declare Multidimensional Array in Java

```
dataType[][] arrayRefVar; (or)
dataType [][]arrayRefVar; (or)
dataType arrayRefVar[][]; (or)
dataType []arrayRefVar[];
```

Example to instantiate Multidimensional Array in Java

```
int[][] arr=new int[3][3];//3 row and 3 column
```

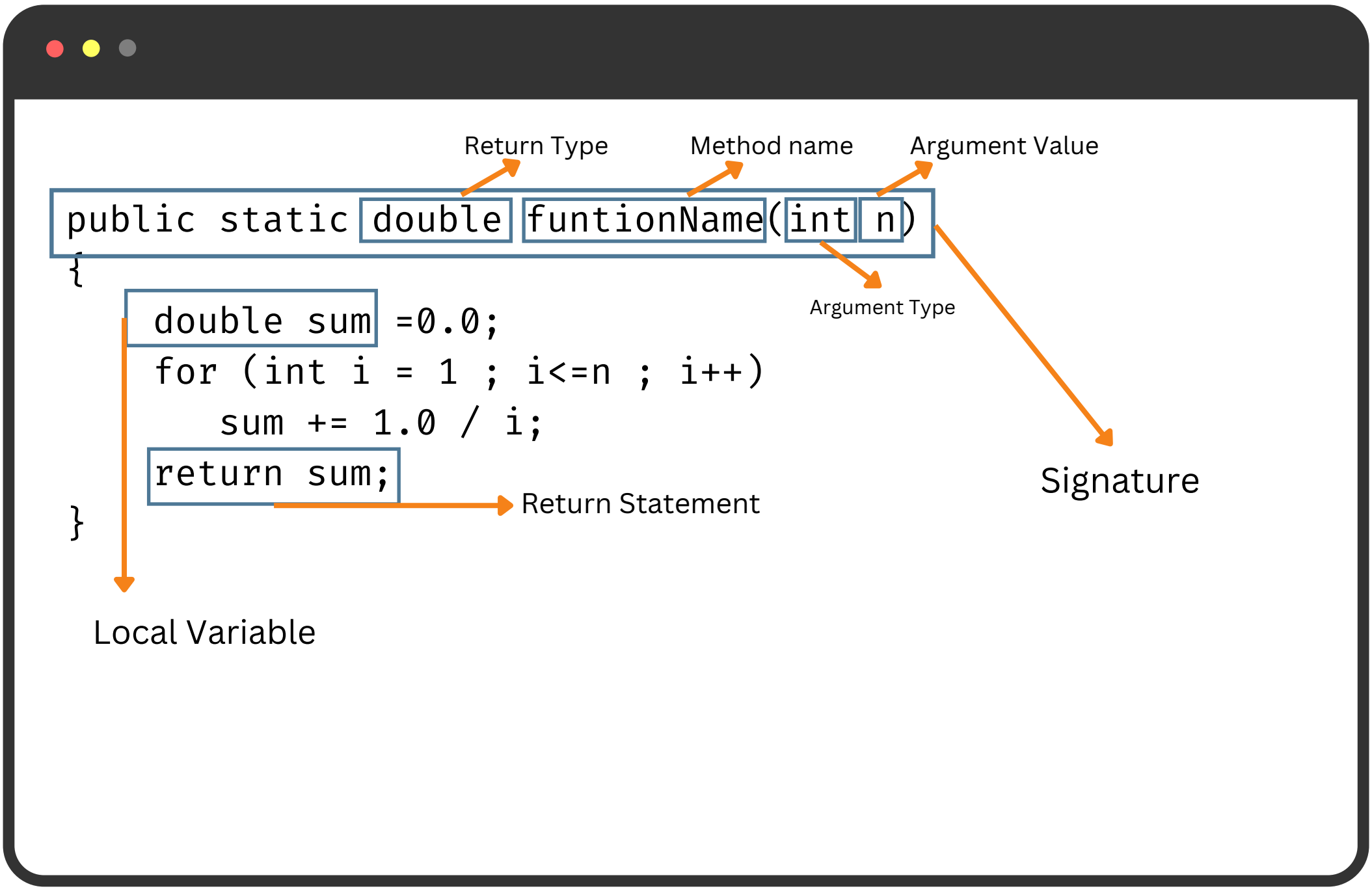
Arrays

Multidimensional Array in Java

Example to initialize Multidimensional Array in Java

```
arr[0][0]=1;
arr[0][1]=2;
arr[0][2]=3;
arr[1][0]=4;
arr[1][1]=5;
arr[1][2]=6;
arr[2][0]=7;
arr[2][1]=8;
arr[2][2]=9;
```

Functions


Return Type
Method name
Argument Value
public static double funtionName(int n)
{
double sum =0.0;
for (int i = 1 ; i<=n ; i++)
sum += 1.0 / i;
return sum;
}
Argument Type
Signature
Return Statement
Local Variable

Functions

absolute value of an `int` value

```
public static int abs(int x)
{
   if (x < 0) return -x;
   else       return  x;
}
```

absolute value of a `double` value

```
public static double abs(double x)
{
   if (x < 0.0) return -x;
   else         return  x;
}
```

primality test

```
public static boolean isPrime(int n)
{
   if (n < 2) return false;
   for (int i = 2; i <= n/i; i++)
      if (n % i == 0) return false;
   return true;
}
```

hypotenuse of a right triangle

```
public static double hypotenuse(double a, double b)
{  return Math.sqrt(a*a + b*b);  }
```

harmonic number

```
public static double harmonic(int n)
{
   double sum = 0.0;
   for (int i = 1; i <= n; i++)
      sum += 1.0 / i;
   return sum;
}
```

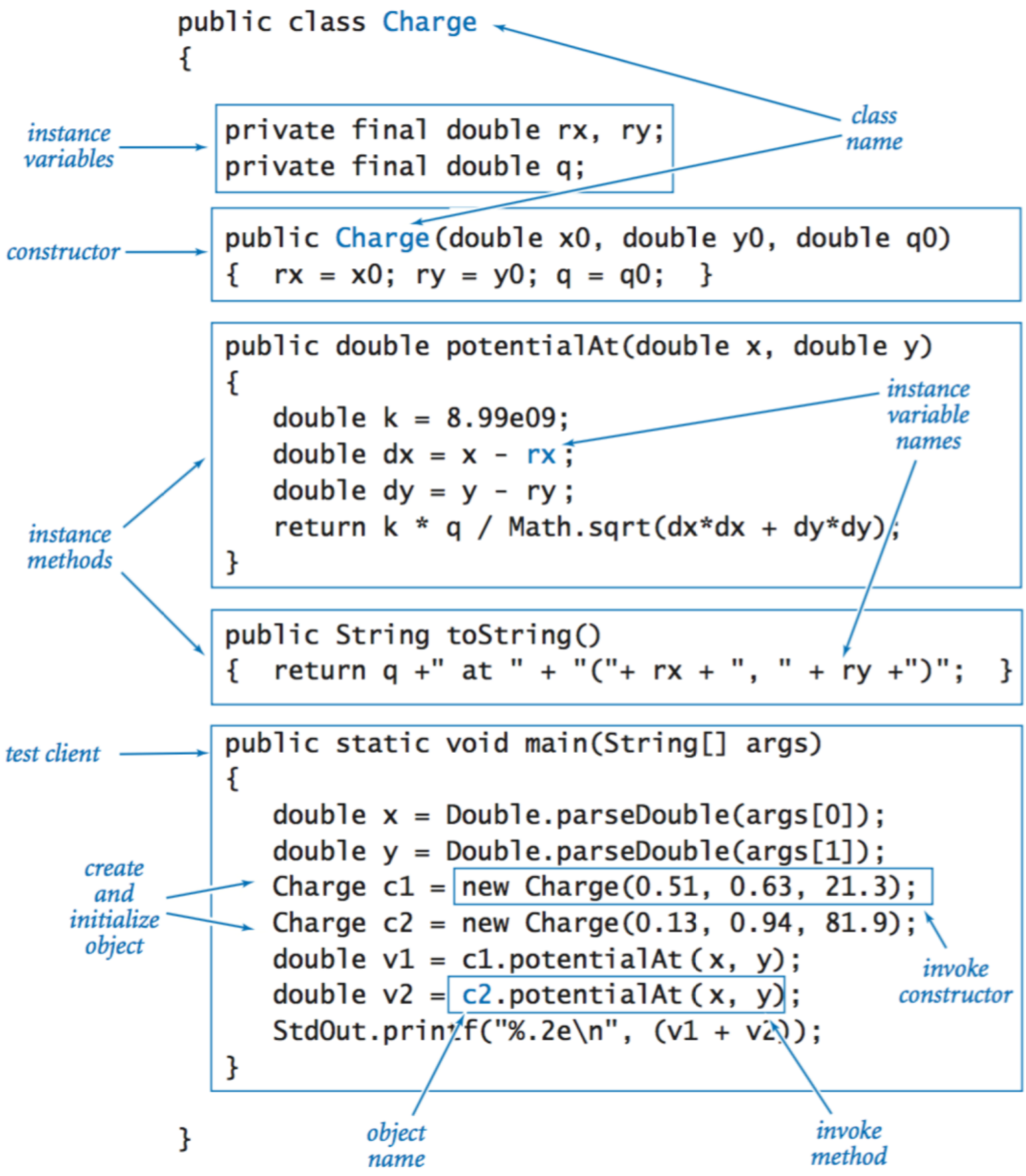
uniform random integer in [0, n)

```
public static int uniform(int n)
{  return (int) (Math.random() * n);  }
```

draw a triangle

```
public static void drawTriangle(double x0, double y0,
                                double x1, double y1,
                                double x2, double y2 )
{
   StdDraw.line(x0, y0, x1, y1);
   StdDraw.line(x1, y1, x2, y2);
   StdDraw.line(x2, y2, x0, y0);
}
```

Classes


public class Charge
{
private final double rx, ry;
private final double q;
public Charge(double x0, double y0, double q0)
{ rx = x0; ry = y0; q = q0; }
public double potentialAt(double x, double y)
{
double k = 8.99e09;
double dx = x - rx;
double dy = y - ry;
return k * q / Math.sqrt(dx*dx + dy*dy);
}
public String toString()
{ return q +" at " + "("+ rx + ", " + ry +")"; }
public static void main(String[] args)
{
double x = Double.parseDouble(args[0]);
double y = Double.parseDouble(args[1]);
Charge c1 = new Charge(0.51, 0.63, 21.3);
Charge c2 = new Charge(0.13, 0.94, 81.9);
double v1 = c1.potentialAt(x, y);
double v2 = c2.potentialAt(x, y);
StdOut.printf("%.2e\n", (v1 + v2));
}
}
instance variables
class name
constructor
instance variable names
instance methods
test client
create and initialize object
invoke constructor
object name
invoke method