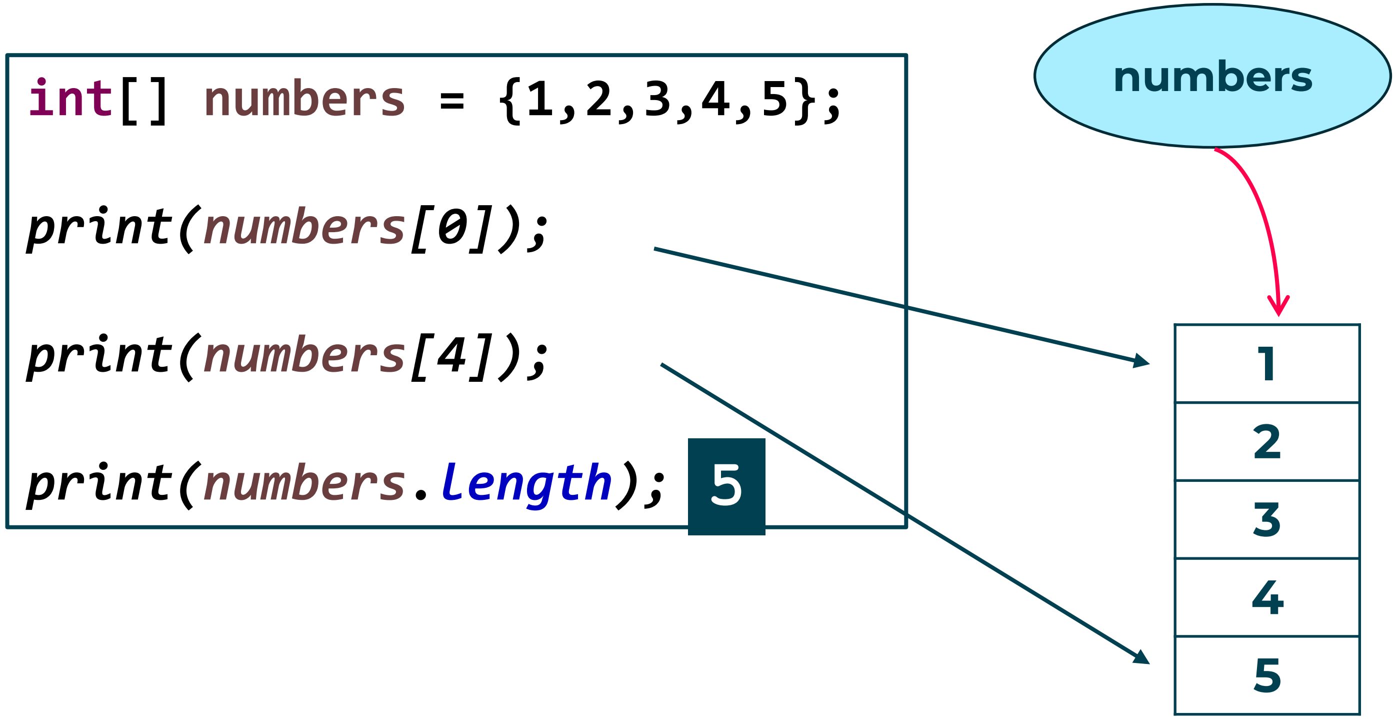
Loops
Introduction to Arrays

CONTENTS

- **Objectives**
  - To cover Java's looping constructs
- **Contents**
  - Looping control statements
    - while and for loop examples
  - Nested loops + QUIZ
  - Exiting and branching options
  - The enhanced for loop (for 'each')
  - Choosing which loop construct to use
- **Hands on Labs**

Introduction to Arrays

- **Loops are often used with arrays and collection.**
- **You'll see a deeper discussion of arrays later on**
- **Array is a fixed-size collection of elements of the same** data type



```
int[] numbers = {1,2,3,4,5};

print(numbers[0]);

print(numbers[4]);

print(numbers.length);
```

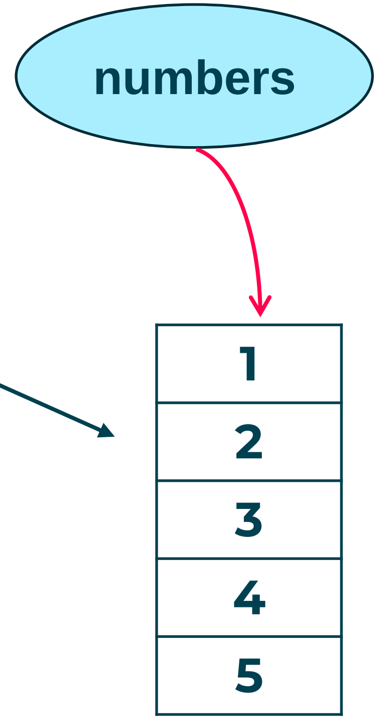
```
5
```

Introduction to Arrays

- **You can change array elements**
- **Appending, Inserting and removing an element is hard**

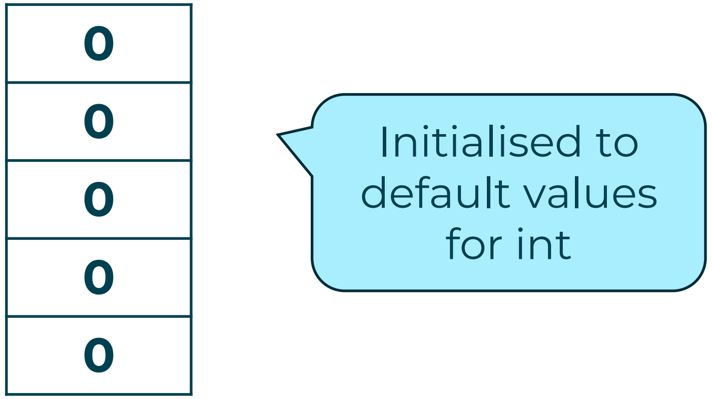
```
int[] numbers = {1,2,3,4,5};

numbers[1] = 22;

print(numbers[1]);
```

```
22
```



Create an empty array

```
int[] numbers = new int[5];
```



0

0

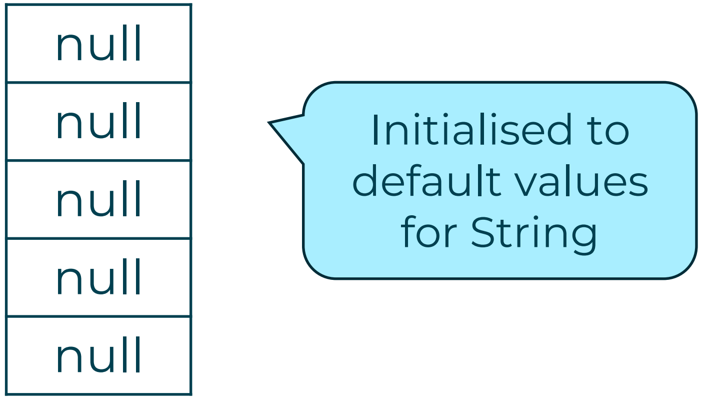
0

0

0

Initialised to default values for int

```
String[] names = new String[5];
```



null

null

null

null

null

Initialised to default values for String

Iteration using while and for Statements

```
while ( boolean_expression ) {
  statement(s);
}
```

```
do {
  statement(s);
} while ( boolean_expression );
```

```
for ( init_expr; boolean_expr; update_expr ) {
  statement(s);
}
```

We'll see iteration using enhanced for loops later on in this chapter

Iteration using while Loops

Declare/Init

Exit condition

```
int i = 0;
while (i < 10)
{
  System.out.printf("%d squared = %d\n", i, i*i );
  i++;
}
```

Changing value

- or put the test condition at the end of the loop

Do these 2 fragments achieve the same result?

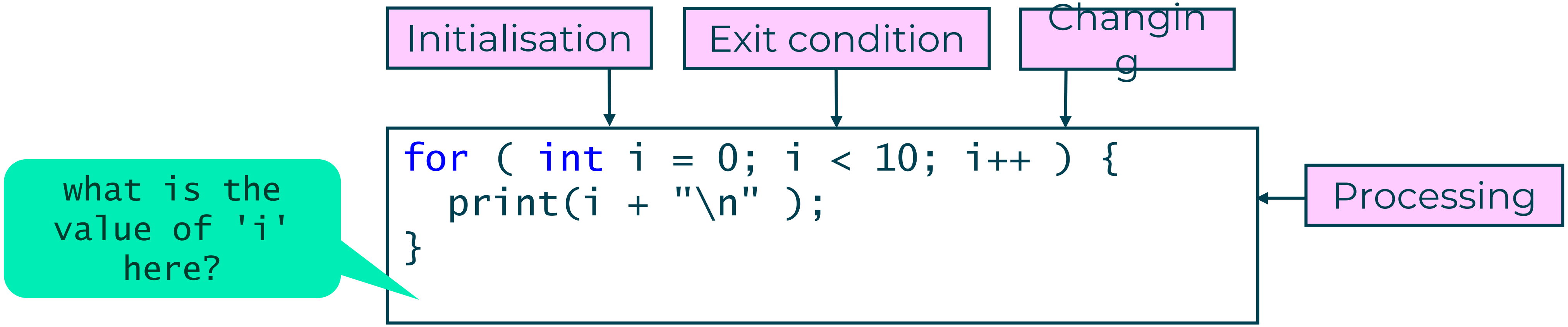
```
int i = 0;
do
{
  System.out.printf("%d squared = %d\n", i, i*i );
  i++;
} while (i < 10);
```

what is the value of 'i' here?

The for loop

Initialisation can include a declaration

- Declared variable is in scope only inside the loop



```
for ( int i = 0; i < 10; i++ ) {
  print(i + "\n" );
}
```

what is the value of 'i' here?

Initialisation and update can be a list of ',' separated expressions

```
for( int i = 0, j = 10; i < j; i++, j-- ) {
       print( i * j + "\n");
}
```

Nested loops Quiz

```
for( int i = 0; i < 3; i++ )
{
    // Statements(A)
  for( int j = 0; j < 5; j++ )
  {
    // Statements(B)
  }
  // Statements(C)
}
// Statements(D)
```

Quiz

1. What would be the values of i & j the 1st time Statements(B) runs?
2. What would be the values of i & j the 3rd time Statements(B) runs?
3. What are the next 2 things that happen after Statements(C)?
4. What is the value of j the last time Statements(A) runs?
5. What is the value of j the last time Statements(C) runs?
6. What is the largest value (i * j) ever reaches?

Using break to exit a any loop

```
double money = 50;          // £ pounds
double interest = 0.06;     // percent

for (int years = 1; money < 1000; years++) {
     money += (money * interest);
     print("Year : "+ years + ": " + money);

     double tax = money * 0.40;
     if (tax > 100) {
          print("Tax is > 100");
          break;
     }
}
```

Break out of the current loop

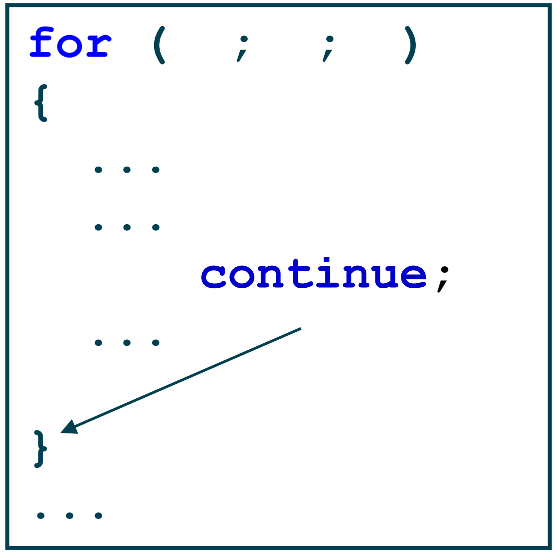
```
Year 1: 53
Year 2: 56
Year 3: 59
Year 4: 63
Year 5: 66
Year 6: 70
Year 7: 75
Year 8: 79
Year 9: 84
Year 10: 89
Year 11: 94
Year 12: 100
Year 13: 106
Year 14: 113
Year 15: 119
Year 16: 127
Year 17: 134
Year 18: 142
Year 19: 151
Year 20: 160
Year 21: 169
Year 22: 180
Year 23: 190
Year 24: 202
Year 25: 214
Year 26: 227
Year 27: 241
Year 28: 255
Tax is > 100
```

Java

```
outer_loop:
for (  ;  ;  ) {
 for (  ;  ;  ) {
  if (...) {
    break outer_loop;
  }
 }
}
..
```

Break out of nested loops

continue



```
for (   ;   ;   )
{
   ...
   ...
        continue;
   ...

}
...
```

```
for ( int i = 0; i < 10; i++ ) {
  if ( i % 4 == 0) {
     // few statements
     continue;
  }
  // many statements
}
```

```
for ( int i = 0; i < 10; i++ ) {
  if ( i % 4 == 0) {
    // few statements
  }
  else {
    // many statements
  }
  // no code here!!
}
```

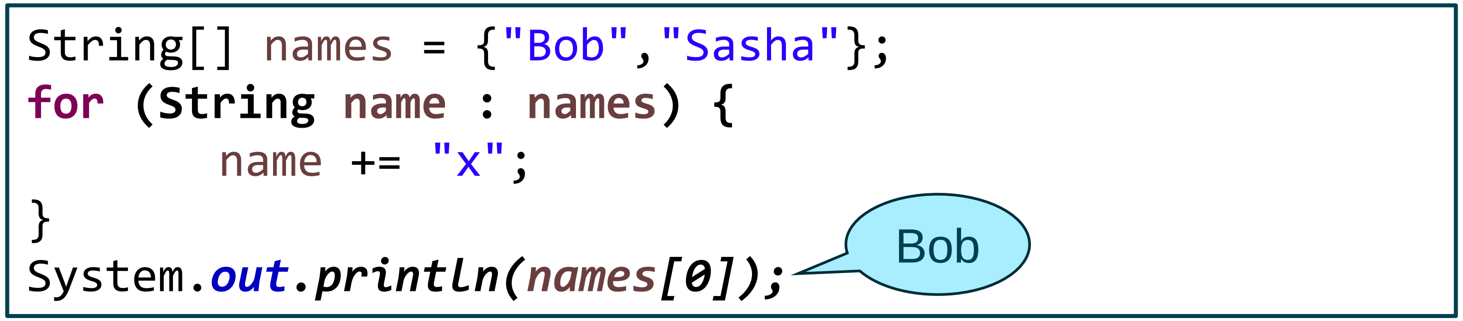
Can be coded as

Java: The enhanced for loop

For iterating over a collection or an array without testing for the bounds

```java
public void processNames( String[] names ) {
  for (String name : names ) {
    System.out.println( name );
  }
}
```

Read as: foreach String ‘name’ in the ‘names’ collection



```java
String[] names = {"Bob","Sasha"};
for (String name : names) {
        name += "x";
}
System.out.println(names[0]);
```

Bob

The elements are considered read-only

An example for an enhanced for loop

```
int[] numbers = {1,2,3,4,5};

int i = 0;
While(i < numbers.length) {
        print(numbers[i]);
        i++;
}
```

```
for (int i = 0; i < numbers.length; i++) {
        print(numbers[i]);
}
```

```
for(int no : numbers) {
        print(no);
}
```

Java

Review

- `while` loop
  - `do while` variation
- `for` loop
- Nested loops
- Branching options
- Enhanced `for` loop – known as ‘foreach’ loop

Hands On Labs

- Calculate grades for a group of students
- How long does it take to double your money?