Chapter 11 Keywords in Java

1. What is Keywords?

A Keywords are special reserved word in java that have a fixed meaning and cannot be used as names for variables, classes or methods.

Key Points:

- There are 50 + keywords in java.
- They are case- sensitive. (class ≠ Class)
- Cannot be used as identifiers. (Variable, class, method names)

Keywords:

Keyword	Usage
abstract	It helps to declare an abstract class or method. This means it has no implementation and must be overridden by a subclass.
assert	This is used for debugging to test a boolean condition and throws AssertionError in false condition.
boolean	It is a comparatively old data type that can hold true or false values.
break	This keyword is used to exit a loop (for, while, do-while) or a switch statement.
byte	It is an old data type that can hold an 8-bit signed integer.
case	It can switch a statement to define different blocks of code to be executed based on a matching value.
catch	Use it with try to catch and handle exceptions that occur within the try block.
char	It is also a data type that can hold a 16-bit Unicode character.
class	It can declare a class, which is a blueprint for creating objects.
const (reserved but unused)	It was used for constants but final is now used for that purpose.
continue	It can skip the rest of the current iteration of a loop and proceeds to the next iteration.
default	Use it in switch statements to specify a block of code to be executed if no case matches. You can also use it in interfaces for default methods.
do	Use it in conjunction with while to create a do-while loop. It then executes at least once before ensuring the condition.
double	It is also an old data type that can hold a 64-bit double-precision floating-point number.
else	Use it with an if statement to specify a block of code to be executed if the if condition is false.
enum	It can declare an enumerated type, a special class that represents a group of named constants.
extends	It can indicate that a class inherits from another class or an interface extends another interface.
final	Use it to declare a variable as a constant (cannot be reassigned), a method as non-overridable or a class as non-inheritable.
finally	Use it with try-catch to define a block of code that will always be executed, without depending on exceptions.

float	It is an old data type that can hold a 32-bit single-precision floating-point number.
for	Use it to create a for loop for iterating a specific number of times or over collections.
goto (reserved but unused)	It allows transferring control to a labeled statement, but its use is generally discouraged in structured programming.
if	It creates a conditional statement that executes a block of code if a specified condition is true.
implements	Use it to specify that a class implements one or more interfaces that provide implementations for their abstract methods.
import	Use it to import classes, interfaces or complete packages from other libraries or parts of the codebase.
instanceof	It is a binary operator that tests whether an object is an instance of a specific class or implements a particular interface.
int	It is an old data type that can hold a 32-bit signed integer.
interface	It can declare an interface, a collection of abstract methods and constants that a class can implement.
long	Use this old data type to hold a 64-bit signed integer.
native	Use it to declare a method whose implementation is provided by platform-specific code (e.g., C/C++).
new	Use it to create new objects (instances of a class) or arrays.
package	It can declare a package, which organizes classes and interfaces into a hierarchical structure.
private	Use it to access modifiers that restrict access to a member (field, method, constructor) to within its own class.
protected	It is used to access modifiers that allow access to a member within its own class and their subclasses.
public	Another access modifier that allows access to a member from any class.
return	Use it to exit a method and optionally return a value.
short	It is another old data type that can hold a 16-bit signed integer.
static	It can declare a member (field, method, nested class) that belongs to the class itself, rather than to any specific object.
strictfp	Use it to ensure strict floating-point computations that will make results consistent across different platforms.
super	It refers to the immediate superclass of the current class used to access superclass members or constructors.
switch	It is a control flow statement that allows a variable to be tested for equality against a list of values.
synchronized	Use it for multithreading to ensure that a block of code or a method can only be executed by one thread at a time.
this	It refers to the current object.
throw	Use it to explicitly throw an exception.
throws	It is used in a method declaration to indicate that the method might throw certain types of exceptions.

transient	It can mark a field to indicate that it should not be serialized when an object is written to a persistent storage.
try	It can define a block of code where exceptions might occur. It is often followed by catch and/or finally blocks.
void	It is a method that does not return any value.
volatile	It can declare a field to indicate that its value may be modified asynchronously by multiple threads that ensure visibility of changes.
while	It builds a while loop that repeatedly executes a block of code as long as a condition is true.
true	A boolean literal representing the logical truth value.
false	A boolean literal representing the logical false value.
null	A literal representing the absence of an object reference.
_ (Underscore)	Reserved for future use.

Common Keywords:

1. int – Declares a number variable (e.g... int age = 10;)
2. class – Declares a class (e.g.... class Student {})
3. void – No return value (e.g.... void show () {})
4. if, else – Conditional branching (e.g.... (if age > 10)) else {}
5. for, while – Loops (e.g.... (for int i = 0; i <=n; i++) (while (i < 5))
6. return – Return a value (e.g. return a)
7. new – Create new objects (e.g. new scanner (System.in))
8. this – Refers to the current object. (e.g. this.name = name;)
9. static – Belongs to the class not from object (e.g. static int count;)
10. super – Refer to the parent class (e.g. super () ;)
11. final – Make variable / methods / class unchangeable (e.g. final int x = 5;)

Static Keyword:

Static keyword is used to define variables or methods that belongs to the class, not object.

Example:

```
class Counter {
  static int count = 0;
  Counter() {
    count++;
    System.out.println("Object count = " + count);
  }
  public static void main(String[] args) {
    new Counter();
    new Counter();
  }
}
```

this keyword:

The 'this' keyword refers to the current object of the class.

Example:

```
class Student {
  String name;
  Student(String name) {
    this.name = name;
  }
  void show() {
```

```
    System.out.println("My name is " + name);
  }
  public static void main(String[] args) {
    Student s = new Student("Riya");
    s.show();
  }
}
```

super keyword:

The super keyword is used to access the parent (superclass) from a child (subclass).

Example:

```
class Animal {
  Animal() {
    System.out.println("I am an Animal");
  }
}
class Dog extends Animal {
  Dog() {
    super();  // calls Animal's constructor
    System.out.println("I am a Dog");
  }
  public static void main(String[] args) {
    new Dog();
  }
}
```

final keyword:

The final keyword is used to make something unchangeable.
You can use with Variable, method, class

Example:

```
class School {
  final String schoolName = "Sunshine School";

  void show() {
    System.out.println("School: " + schoolName);
  }

  public static void main(String[] args) {
    School s = new School();
    s.show();
  }
}
```

Exercises

1. Multiple Choice Questions (MCQs)

1. What is a keyword?
   - a) A class
   - b) A special reserved word
   - c) A loop
   - d) None
2. Can we use keywords as variable names?
   - a) Yes
   - b) No
3. Which keyword is used to create an object?
   - a) class
   - b) new
   - c) final
   - d) super
4. which keyword refer current object?
   - a) this
   - b) super
   - c) static
   - d) new
5. Which keyword call parent element?
   - a) this
   - b) class
   - c) new
   - d) super

2. Fill in the Blanks

1. The keyword refers to parent class _____
2. The keyword _____used to create new object
3. The_____keyword makes a variable or method unchangeable
4. The _____ is used to define a class in java
5. The _____ keyword refer to the current object

3. true or False

1. The final keyword allows a variable to be changed later.
2. This keyword refers to the current object.
3. The super keyword is used to access methods and constructors from the parent class.
4. You can use int as a variable name in Java.
5. The static keyword means the variable or method belongs to the class, not an object.

Home task

Create a new project in BlueJ.

1. Create a class called Visitor where each time an object is created, a static variable totalVisitors increases by 2.
2. Create a class Car with a constructor that takes a modelName as input.
3. Create two classes: Vehicle and Bike.
   The Bike class should extend the Vehicle class and call the parent constructor using the super keyword.
4. Create a class Library that has a final variable libraryName and print it in a method.