for data types

|  |  |
| --- | --- |
| byte | 1 byte Stores whole numbers from -128 to 127 |
| short | 2 bytes Stores whole numbers from -32,768 to 32,767 |
| int | 4 bytes Stores whole numbers from -2,147,483,648 to 2,147,483,647 |
| long | 8 bytes Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |
| float | 4 bytes Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits |
| double | 8 bytes Stores fractional numbers. Sufficient for storing 15 decimal digits |
| char | 2 bytes Stores a single character/letter or ASCII values |
| boolean | 1 bit Stores true or false values |

flow control

|  |  |
| --- | --- |
| if | The Java if statement is used to test the condition. It checks boolean condition: true or false. There are various types of if statement in Java. |
| else | The Java if-else statement also tests the condition. It executes the if block if condition is true otherwise else block is executed. |
| switch | The Java switch statement executes one statement from multiple conditions. It is like if-else-if ladder statement. The switch statement works with byte, short, int, long, enum types, String and some wrapper types like Byte, Short, Int, and Long. Since Java 7, you can use strings in the switch statement. |
| case | The value of the expression is compared with the values of each case. If there is a match, the associated block of code is executed. |
| default | The default keyword specifies some code to run if there is no case match |
| for | The Java for loop is used to iterate a part of the program several times. If the number of iteration is fixed, it is recommended to use for loop. |
| do | The Java do-while loop is used to iterate a part of the program repeatedly, until the specified condition is true. If the number of iteration is not fixed and you must have to execute the loop at least once, it is recommended to use a do-while loop.  Java do-while loop is called an exit control loop. Therefore, unlike while loop and for loop, the do-while check the condition at the end of loop body. The Java do-while loop is executed at least once because condition is checked after loop body. |
| while | The Java while loop is used to iterate a part of the program repeatedly until the specified Boolean condition is true. As soon as the Boolean condition becomes false, the loop automatically stops.  The while loop is considered as a repeating if statement. If the number of iteration is not fixed, it is recommended to use the while loop. |
| break | When a break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.  The Java break statement is used to break loop or switch statement. It breaks the current flow of the program at specified condition. In case of inner loop, it breaks only inner loop.  We can use Java break statement in all types of loops such as for loop, while loop and do-while loop. |
| continue | The continue statement is used in loop control structure when you need to jump to the next iteration of the loop immediately. It can be used with for loop or while loop.  The Java continue statement is used to continue the loop. It continues the current flow of the program and skips the remaining code at the specified condition. In case of an inner loop, it continues the inner loop only.  We can use Java continue statement in all types of loops such as for loop, while loop and do-while loop. |
| return | A return statement causes the program control to transfer back to the caller of a method. Every method in Java is declared with a return type and it is mandatory for all java methods. A return type may be a primitive type like int, float, double, a reference type or void type(returns nothing). |

modifiers

|  |  |
| --- | --- |
| public | public is a Java keyword which declares a member's access as public. Public members are visible to all other classes. This means that any other class can access a public field or method. Further, other classes can modify public fields unless the field is declared as final. |
| private | The private keyword is an access modifier used for attributes, methods and constructors, making them only accessible within the declared class. |
| protected | The protected keyword is an access modifier used for attributes, methods and constructors, making them accessible in the same package and subclasses. |
| Static | The static keyword in Java is mainly used for memory management. The static keyword in Java is used to share the same variable or method of a given class. The users can apply static keywords with variables, methods, blocks, and nested classes. The static keyword belongs to the class than an instance of the class. |
| final | When a variable is declared with final keyword, its value can’t be modified, essentially, a constant. This also means that you must initialize a final variable. If the final variable is a reference, this means that the variable cannot be re-bound to reference another object, but the internal state of the object pointed by that reference variable can be changed i.e. you can add or remove elements from the final array or final collection.  When a class is declared with final keyword, it is called a final class. A final class cannot be extended(inherited).  When a method is declared with final keyword, it is called a final method. A final method cannot be overridden. The Object class does this—a number of its methods are final. We must declare methods with the final keyword for which we are required to follow the same implementation throughout all the derived classes. |
| abstract | The class cannot be used to create objects (To access an abstract class, it must be inherited from another class. You will learn more about inheritance and abstraction in the Inheritance and Abstraction chapters) |
| synchronized | Methods can only be accessed by one thread at a time |
| native | The native keyword in Java is applied to a method to indicate that the method is implemented in native code using JNI (Java Native Interface). The native keyword is a modifier that is applicable only for methods, and we can’t apply it anywhere else. The methods which are implemented in C, C++ are called native methods or foreign methods. |
| strictfp | strictfp is a keyword in java used for restricting floating-point calculations and ensuring same result on every platform while performing operations in the floating-point variable. |
| transient | Attributes and methods are skipped when serializing the object containing them |
| volatile | The value of an attribute is not cached thread-locally, and is always read from the "main memory" |

exception handling

|  |  |
| --- | --- |
| try | Java try block is used to enclose the code that might throw an exception. It must be used within the method.  If an exception occurs at the particular statement in the try block, the rest of the block code will not execute. So, it is recommended not to keep the code in try block that will not throw an exception.  Java try block must be followed by either catch or finally block. |
| catch | Java catch block is used to handle the Exception by declaring the type of exception within the parameter. The declared exception must be the parent class exception ( i.e., Exception) or the generated exception type. However, the good approach is to declare the generated type of exception.  The catch block must be used after the try block only. You can use multiple catch block with a single try block. |
| finally | ava finally block is a block used to execute important code such as closing the connection, etc.  Java finally block is always executed whether an exception is handled or not. Therefore, it contains all the necessary statements that need to be printed regardless of the exception occurs or not.  The finally block follows the try-catch block. |
| throw | The Java throw keyword is used to throw an exception explicitly.  We specify the exception object which is to be thrown. The Exception has some message with it that provides the error description. These exceptions may be related to user inputs, server, etc.  We can throw either checked or unchecked exceptions in Java by throw keyword. It is mainly used to throw a custom exception. We will discuss custom exceptions later in this section.  We can also define our own set of conditions and throw an exception explicitly using throw keyword. For example, we can throw ArithmeticException if we divide a number by another number. Here, we just need to set the condition and throw exception using throw keyword. |
| throws | The Java throws keyword is used to declare an exception. It gives an information to the programmer that there may occur an exception. So, it is better for the programmer to provide the exception handling code so that the normal flow of the program can be maintained.  Exception Handling is mainly used to handle the checked exceptions. If there occurs any unchecked exception such as NullPointerException, it is programmers' fault that he is not checking the code before it being used. |
| assert | The Java assert keyword allows developers to quickly verify certain assumptions or state of a program.  The Java assert keyword has been available for many years but remains a little-known feature of the language. It can help remove lots of boilerplate code, make the code more readable, and help identify bugs early in program development.  Just remember that assertions aren't enabled by default, so never assume they will be executed when used in the code. |

class related

|  |  |
| --- | --- |
| class | The class keyword is used to create a class. Every line of code that runs in Java must be inside a class. A class should always start with an uppercase first letter, and the name of the java file must match the class name. |
| package | Package in Java is a mechanism to encapsulate a group of classes, sub packages and interfaces. |
| import | import is a Java keyword.  It declares a Java class to use in the code below the import statement. Once a Java class is declared, then the class name can be used in the code without specifying the package the class belongs to.  Use the '\*' character to declare all the classes belonging to the package. |
| extends | The extends keyword extends a class (indicates that a class is inherited from another class).  In Java, it is possible to inherit attributes and methods from one class to another. |
| implements | An interface is an abstract "class" that is used to group related methods with "empty" bodies:  To access the interface methods, the interface must be "implemented" (kinda like inherited) by another class with the implements keyword (instead of extends). The body of the interface method is provided by the "implement" class. |
| interface | An interface is an abstract "class" that is used to group related methods with "empty" bodies:  To access the interface methods, the interface must be "implemented" (kinda like inherited) by another class with the implements keyword (instead of extends). The body of the interface method is provided by the "implement" class. |

object related keywords

|  |  |
| --- | --- |
| new | The new keyword creates new objects. |
| instanceof | The instanceof keyword checks whether an object is an instance of a specific class or an interface.  The instanceof keyword compares the instance with type. The return value is either true or false. |
| super | The super keyword refers to superclass (parent) objects.  It is used to call superclass methods, and to access the superclass constructor.  The most common use of the super keyword is to eliminate the confusion between superclasses and subclasses that have methods with the same name. |
| this | The this keyword refers to the current object in a method or constructor.  The most common use of the this keyword is to eliminate the confusion between class attributes and parameters with the same name (because a class attribute is shadowed by a method or constructor parameter). |