1) What is a keyword in Java?

A keyword in Java is a word that already has a fixed meaning in the Java language. Java understands these words and uses them to perform specific actions. Because Java already knows these words, we are not allowed to use them for our own names. Keywords are always written in small letters. Examples are int, class, if, else, public. Since Java uses these words internally, we cannot use them as names for variables, methods, or classes.

2) What is an identifier in Java?

An identifier is a name that we give to something in our program so that Java can identify it. These names are created by programmers. For example, when we name a variable, a method, or a class, that name is called an identifier. Names like studentName, calculateTotal, or Employee are identifiers. In short, identifiers are user-defined names used to identify program elements.

3) Where can we not use keywords in a program?

We cannot use keywords as names for variables, methods, classes, or any other identifiers. For example, we cannot create a variable named int or a class named class because these are keywords and Java has already reserved them for its own use.

4) What are the rules for identifiers?

An identifier can contain letters, numbers, and the special symbols underscore (_) and dollar sign ($).
An identifier must not start with a number. It should start with a letter, underscore, or dollar sign.
Keywords cannot be used as identifiers.
Spaces are not allowed in identifiers.
Identifiers are case-sensitive, which means name and Name are treated as different.

Think of it like naming people. Keywords are reserved names that already belong to Java, and identifiers are the names we are allowed to give to our own variables, methods, and classes by following some basic rules.
5) What is a magic value (magic number) in Java?

A magic value in Java means using a number directly in the code without explaining what it is for. When someone else reads the code later, they may not understand why that number is used. This makes the code confusing and hard to maintain.

For example, if you write a condition like salary > 50000, the number 50000 appears suddenly and there is no clue why this value is important. A new person reading the code will not know its purpose.

Magic values are not good practice because if the value needs to change, you have to search and update it in many places. Instead, we should store that value in a variable or constant with a meaningful name so the code becomes easy to read and understand.

In simple words, a magic value is a hard-coded number in the program that has no clear meaning when someone reads the code.