Primitive Data Types: Primitive data types represent basic values and are built into the Java language. They are simple and fundamental, representing single values. Java has 8 primitive data types:

1. byte: Represents a 8-bit signed integer value.
2. short: Represents a 16-bit signed integer value.
3. int: Represents a 32-bit signed integer value.
4. long: Represents a 64-bit signed integer value.
5. float: Represents a 32-bit floating-point value.
6. double: Represents a 64-bit floating-point value.
7. char: Represents a 16-bit Unicode character.
8. boolean: Represents a true or false value.

Primitive data types are stored directly in memory and are usually more memory-efficient than non-primitive data types. They have a fixed size and are faster to access.

Non-Primitive (Reference) Data Types: Non-primitive data types are also known as reference data types because they store references (memory addresses) to objects rather than the actual values themselves. These data types are used to work with complex data structures and objects. Some common examples of non-primitive data types in Java include:

1. Classes: Blueprint for creating objects, containing attributes (variables) and methods (functions).
2. Interfaces: A contract that defines a set of methods that a class must implement.
3. Arrays: Ordered collection of elements of the same type.
4. Enums: A special data type that defines a set of constant values.

Non-primitive data types are more flexible as they can represent complex structures and hold multiple values. However, they tend to consume more memory than primitive data types, as they not only store the actual data but also meta data and references.