1. Statistically Typed Language

   Data types will be checked while compiling. These types of languages are used heavily in enterprise application development.

   Dynamically Typed Language

   Data types are checked at runtime. This type is not used for enterprise application development.

   Strongly Typed Language

   Data types are defined strictly. These types of languages are used heavily in enterprise application development.

   Loosely Typed language

   Data types aren't defined strictly.

   Java contains both Statistically typed and Dynamically typed. And it is a strongly typed programming language.

2. Case Sensitive

   Here, the difference between upper & lower case letters is significant. C, C++, Java, C#, Python fall under this category.

   Case Insensitive

   These type of languages do not distinguish between upper & lower case letters. SQL, Pascal fall under this category.

   Case Sensitive Insensitive

   These are a hybrid type language. Some of the identifiers of these languages are case-sensitive while others are case insensitive. JavaScript falls under this category.

   Java is a case-sensitive programming language.

3. Identity Conversion

   Identity conversion occurs when two objects of the same type are assigned to each other. This is an implicit conversion. This can be performed between objects of the same type and this conversion won't change the value of the object.

```
Int numberOne = 10;
int numberTwo = numberOne;
```

```
String stringOne = "Hello";
String stringTwo = StringOne;
```

4. Primitive Widening Conversion

   This is a implicit type conversion that happens when a smaller data type is converted into a larger data type. No data will be lost.

   byte myByte = 10;

   Int myInt = myByte;

   byte → short → int → long

5. Run-time constants & Compile-time constants

   In compile-time constants, compiler identifies the value while compiling and in run-time constants compiler cannot identify the value while compiling. The program should run to identify the value.

   Final int ctConstants =10; //is a compile time constant

   int rtConstants =10; //is a run time constant

6. Narrowing primitive conversion

   Implicit narrowing primitive conversions are performed automatically by the compiler when a value of a wider primitive type is assigned to a variable of a narrower primitive type. The value of the wider primitive type must be representable in the narrower primitive type.

   Byte myByte =10;

   int myInt = myByte;

   Explicit narrowing conversions are performed explicitly by the programmer using a cast operator.

   Byte myByte =10;

   int myInt = (int)myByte;

7. long → float conversion happens through scientific notation. Accuracy of this conversion will be low. Float data type can store larger numbers than long data type.

8. int is the default data type for integer literals because int data type is large enough to represent most integers that are used in everyday programming and it doesn't take up a lot of memory. Int data type is a good combination of size & precision.

double is the default data type for floating point literals because double has a wider range of precision than float.

9. It happens because byte,char,int & short are the only primitive types that can be represented exactly in each other.

10. short data type can directly be converted into char data type without following the short → int → char route which happens to be the widening & narrowing conversion.