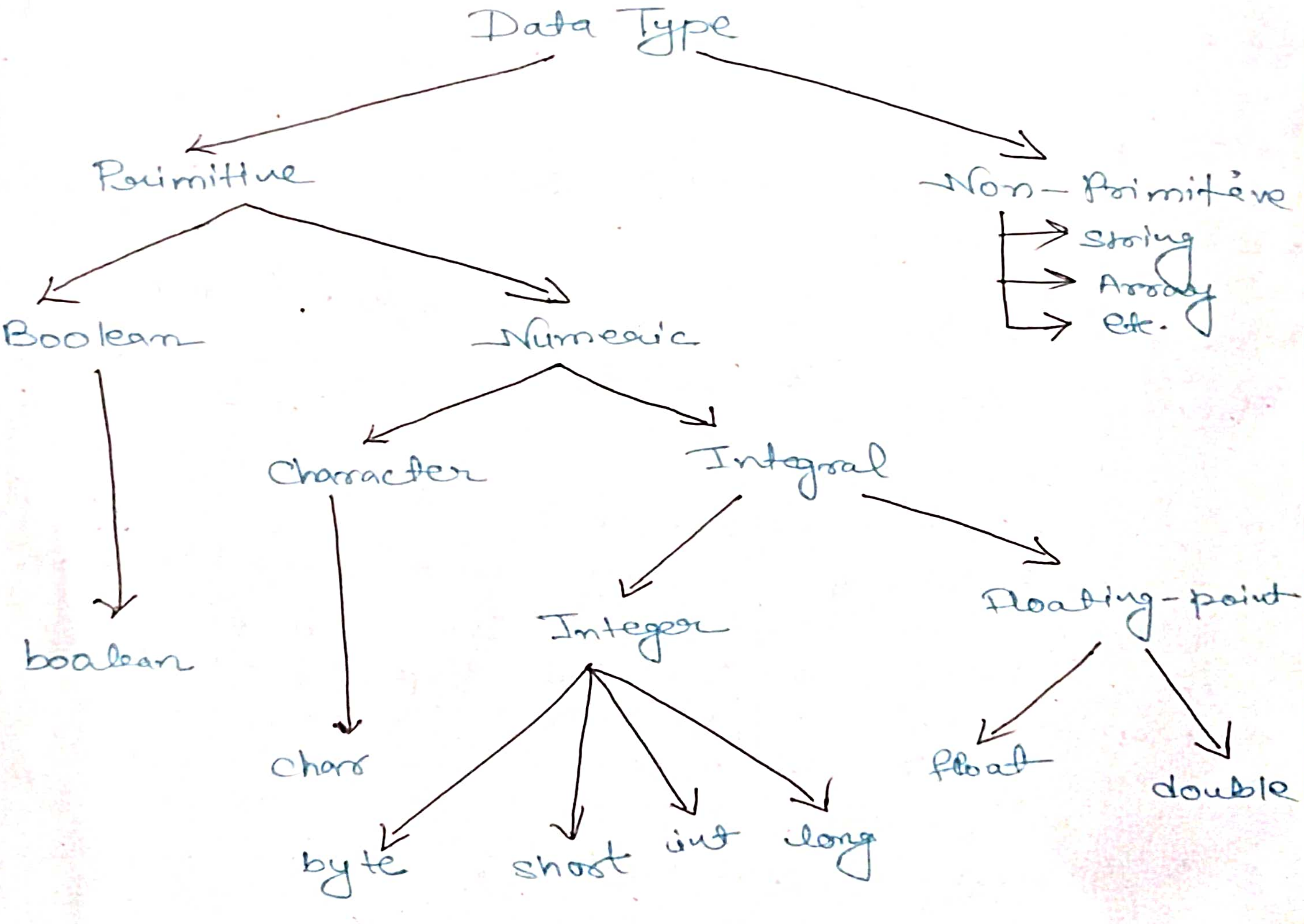
Java Data Types

```
int num = 15;
float f_num = 7.86f;
char letter = 'Y';
boolean bool = false;
String text = "Priya";
```

Data types are divided into two groups:

Primitive data types - includes byte, short, int, long, float, double, boolean and char.

Non-primitive data types - such as String, Arrays and Classes.



Numbers

Primitive number types are divided into two group

Integer types stores whole numbers, positive or negative (such as 123 or -456), without decimals. Valid types are byte short, int and long. Which type you should use, depends on the numeric value.

Floating point types represents numbers with a fractional part, containing one or more decimals. There are two types: float and double.

byte The byte data type can store whole numbers from -128 to 127. This can be used instead of int or other integer types to save memory when you are certain that the value will be within -128 and 127.

```
byte var = 100;
System.out.println(var);
```

short

The short data type can store whole numbers from -32768 to 32767.

```
Short a = 5000;
System.out.println(a);
```

int The int data type can store whole numbers from -2147483648 to 2147483647. In general, and in our tutorial, the int data type is the preferred data type when we create variables with a numeric value.

```
int a = 100000;
System.out.println(a);
```

long The long data type can store whole numbers from -9223372036854775808 to 9223372036854775807. This is used when int is not large enough to store the value. Note that you should end the value with an "L".

```
long a = 15000000000L;
System.out.println(a);
```

Floating Point Types

You should use a floating point type whenever you need a number with a decimal, such as 9.99 or 3.14515.

Float

```
float a = 5.75f;
System.out.println(a);
```

The float data type can store fractional numbers from 3.4e-38 to 3.4e+38. Note that you should end the value with an "f".

double:

The double data type can store fractional numbers from 1.7e-308 to 1.7e+308. Note that you should end the value with a "d".

```
double a = 19.99d;
System.out.println(a);
```

1.7e-308 → 1.7×10^{-308}

1.7 ~~10~~e+308 → 1.7×10^{308}

Use float or double ??

The precision of a floating point value indicate how many digits the value can have after the decimal point. The precision of float is only six or seveen decimal digits, ~~we~~ while double variables have a precision of about 15 digits. Therefore it is safter to use double for most calculations.

e.g. 6/7

float

$$\underline{14.3692356 7898} \leftarrow \text{float} (\times)$$

double (✓)

Scientific Numbers

A floating point number can also be a scientific number with an "e" to indicate the power of 10.

1) 357096 → $\frac{3.57096 \times 10^5}{100000} = 3.57096 \times 10^5.$

2) 124.7 → $\frac{124.7}{1\cancel{0} \cancel{\times} 1000} \times 100\cancel{0}$

$= 1.247 \times 10^2$

3) 0.005600 → $\frac{0.005600}{1000000 \times 1000} \times 1\cancel{000}$

5.600×10^{-3}

4) 0.00098 → $\frac{0.00098}{10.000\cancel{0} \times 10} \times 1\cancel{0}$

$= 9.8 \times 10^{-4}$

Standard Notation	Scientific Notation	E Notation
357096	3.57096×10^5	3.57096e5
124.7	1.247×10^2	1.247e2
0.005600	5.600×10^{-3}	5.600e-3
0.00098	9.8×10^{-4}	9.8e-4

```
float  f1 = 35e3f;  → 35×10^3
double  d1 = 12E4d;  → 12×10^4
System.out.println(f1);
 System.out.println (d1);
```

Booleam

A booleam data type is declared with booleam keyword and can only take the values true or flase. Booleam values are mostly used for conditional testing.

```
boolean  areYouSingle = true;
boolean  doYouHave GirlFriend = false;
System.out.println (areYouSingle); // true
System.out.println (doYouHaveGirlFriend); // false.
```

Characters

The char data type is used to ~~single~~ store a single character. The character must be surrounded by single quotes, like 'A' or 'c'.

```
  char myGrand = 'B';
  System.out.println(myGrade);
```

Alternatively, you can use ASCII values to display certain characters.

```
 char a=65, b=66,  c=67;
System.out.println (a);
System.out.println (b);
 System.out.println(c);
```

String

The String data type is used to store a sequence of characters (text). String values must be surrounded by double quotes.

```
String tutor = "Yash Jain";
  System.out.println(tutor);
```

A string in Java is actually a non-primitive data type, because it refers to an object. The String object has methods that are used to perform certain operations on string.

Non - Primitive Data Types

Non-primitive data types are called reference types because they refer to object.

The main difference between primitive and non-primitive data types are:-

⊙ Primitive types are predefined (already defined) in Java. Non-primitive types are created by the programmer and is not defined by Java (expect for String).

⊙ Non-primitive type can be used to call methods to perform certain operations, while primitive types cannot.

⊙ A primitive type has ~~already~~ always a value, while non-primitive types can be null.

⊙ A primitive type starts with lowercase letter while non-primitive types starts with an uppercase letter. The size of a primitive type depends on the data type, while non-primitive types have all the same size.

⊙ Examples of non-primitive types are Strings, Arrays, Classes, Interface etc.