Unit 4 Lesson 1 VARIABLES AND CONSTANTS + CASTING

In solving a problem we often use variables of different data types. Mathematical calculations often require different type of numbers be stored. For example the result of a division of two integers isn't always an integer, using powers can result in a very large number etc.
To accommodate such cases we can use a process known as "casting" as in casting(assigning) one data type to another.

1. **No loss of data conversions**

The general rule is that up-casting, or assigning a smaller value to a larger capacity variable such as from integer to long is allowed and does NOT require the use of additional functions.
Such conversions are considered safe as there is no loss of information.

Allowed conversions:

From:	**To Any of:**
byte	short, int, long, float, double
short	int, long, float, double
char	int, long, float, double
int	long, float, double
long	float, double
float	double

2. **Conversions with probable loss of data**

When a larger number is assigned to a variable of a smaller capacity such as float to an integer or integer to a byte, the loss of data is likely. To prevent Processing from crashing our program due to this seemingly unreasonable action we need to use explicit casting function. This allows for legitimate transfer of values between data types with or without data loss.

Syntax:
float coefficient =2.75;
int lesser=(int)coefficient;

Cast syntax is to place the type you want to convert to *in brackets **in front of*** the value you want to convert.

This will result in only 2 being stored in **lesser** (Processing truncates, does not round up or down).

Beside the following values write the correct code if incorrect or state "correct".

long num1=989898999; Int num2=num1;	long num1=989898999; int num2=(int)num1;
int num1=200; byte num2=num1;	int num1=200; byte num2=(byte)num1;
long num1=123456789L float num2=num1;	correct
Int x=22334; long y=x;	correct

After completion test your solution in Processing and write the corrections, if any.

3. Constants

Constant are declared when a value may be used repeatedly in a program, however does not need to change. A constant may be sales tax in a program meant for calculating retail prices or a value used in a formula.

```
Syntax:
final float SALES_TAX=13.5;
final String TEACHER=”Ms. Krasteva”;
```

// Note the all caps identifier, typical for constants

Exercise:

1. Beside each declaration state if it is correct or an error. Write in the correction where error was made.

int weight=12.5;	error	float weight=12.5;
double x= 21;	error	int x=21;
char unit=’m^2‘	error	char unit='m^2';
String grade=’A’;	error	char grade='A';
boolean false=yes;	error	boolean yes=false;

2. Consider a program which keeps track of students in a class and their marks. Working in pairs list all variables you think you will need in order to create the program. Declare each variable and beside it write its purpose.

Student variables, using their initials or similar as identifiers for identifiers, with an m beside that to separate this from the other variables. This first variable is for their marks. We may also use m1 and such for more specific marks, not just the average.
ex. for me it'd be
float JeCm=88.75;
I used Je for myself here since JC is a not uncommon initial pair.

However, if this is for more than one year we may want to
int Student1m=88.75;
and so on. This way, we can also do something like
String Student1s="Billy Bob Joe";
with the s standing for student, as in them themselves.