# Critical thinking skills

a)
Legal identifier names are names that follow Java rules (cannot start with numbers, no spaces, no special characters except _ or $, and can't be reserved keywords). Examples of 4 legal identifier names are:
studentName, num1, totalPrice, and my_Age.
All of these follow Java rules — they start with a letter and have no spaces.

b)
Illegal identifier names break rules. Examples:
2num is illegal because identifiers cannot start with a number.
student name is illegal because it has a space.
float is illegal because this is a Java reserved keyword.
price&tax is illegal because symbols like & are not allowed in variable names.

2.

a)
This one needs code because it asks you to write statements.
Two-statement version:

int numBeads;
numBeads = 5;

b)
One-statement version:

int numBeads = 5;

3.

a)
After all the statements run, yourNumber ends up becoming myNumber + 5.
myNumber was 5, so yourNumber = 5 + 5, which becomes 10.
So the final value of yourNumber is 10.

b)

# Critical thinking skills

In this one, the line yourNumber + 7; does nothing because that value is not stored anywhere (no assignment).
myNumber also never gets assigned a starting value — so the final line yourNumber = myNumber; is invalid (unassigned variable).
So this code is incorrect and it would cause a compile error because myNumber has no value before being used.

4.

a) number of basketballs in a store → this is a whole count, so int.
b) price of a basketball → prices use decimals, so double.
c) number of players on a basketball team → whole number count, so int.
d) average age → can be decimals, so double.
e) whether a player has a jersey → two-state yes/no value, so boolean.
f) first initial of a name → single character, so char.

5.

a)
A primitive data type is a basic built-in type like int, double, boolean, or char. It holds a simple value directly.
An abstract data type is more complex — it is not built in and usually represents something that is created or defined by the programmer or defined in a library (like a class). Abstract data types store multiple pieces of data and behaviors together.

b)
A class is the blueprint or definition — it describes what something is and what it can do.
An object is the actual thing created from a class — like a real instance in memory based on that class

11.

a) Since j * k gives a double, and y is an int, we must cast:

y = (int)(j * k);

b) This one does not need casting because z is already a double:

z = j * k;