Worksheet 01 - Operators & Expressions

a) Evaluate each of the following expression below, and indicate its value in the box provided. Be sure to specify the appropriate type of the result (e.g., `7.0` rather than `7` for a `double`, Strings in `"quotes"`, `true` or `false` for a `boolean`).

```
3 + 4 * 5 / 2
```

```
13
```

*NOTE: If expression was 3 + 5 / 2 * 4 , answer would be 11*

```
"over " + 80 + " = old"
```

```
“over 80 = old”
```

```
57 % 10 / 3 + 1.25 * 4
```

```
7.0
```

```
5 + 2 + "(1 + 1)" + 4 + 2 * 3
```

```
“7(1 + 1)46”
```

*NOTE: numeric expression (2*3 first), then from left (5+2=7) until becomes string expression, then remains string expression for each of last 2 + operators (because left operand is always a string)*

b) How many unique items can be encoded (represented) within a computer using each of the following?

a. 1 bit ___ $2^1 = 2$ (bit has 2 states: 0 and 1) ______

b. 4 bits ___ $2^4 = 16$ ________________

c. 5 bits ___ $2^5 = 32$ ________________

d. 2 bytes ___ $2^{16} = 65{,}536$ (char uses 2 bytes = 2*8 bits = 16 bits) _

c) In which of the following answers does the number of bits decrease from most (on the left) to fewest (on the right)?

```
a) byte, long, int, short
b) long, int, byte, char
c) long, int, char, byte
d) int, short, char, byte
```

d) Which of the following is **not** a legal Java statement?

```
a) int w  = (int)  888.8;
b) byte x = (byte) 100L;
c) long y = (byte) 100;    → promoted automatically, so ok
d) byte z = (int) 100L;    → must cast to demote wider type
```

e) Indicate (by circling) which of the following are **valid/invalid** identifiers in Java, and for each identifier, briefly explain why it is valid/not?

a. RESULT valid -> starts with and has no illegal characters

b. 12345 invalid -> starts with a numeric character

c. x12345y valid -> starts with character, can have numeric

d. black&white invalid -> & is an illegal char (& is an operator)

e. answer_7 valid -> '_' is a legal character to have

f) For each of the following expressions, indicate the order in which the operators will be evaluated (by writing a number beneath each operator):

```
a – b – c – d ;
  1   2   3
```

```
a + b / c / d ;
  3   1   2
```

```
a % b / c * d ;
  1   2   3
```

```
(a – ( b – c )) – d ;
   2     1      3
```

```
(a + b) * (c / d) % e ;
   1    3    2    4
```

g) What is the difference between an automatic (promotion) and a manual cast (demotion)? Under what circumstances are automatic casts performed?

- When using an operator, all operands must be converted to same type.
- A value/variable of a given type must match the type it is being assigned to.

Promotion occurs automatically:
- when operands are diff type, both automatically promoted to widest type
- narrower type converted to wider type automatically during assignment

Demotion must be done manually:
- if we want to force an operation to be done on the narrower of two operand's types, or
- if assigning a wider type to a variable of a narrower type