**Exercise 1:**

1. 7
2. Typing in 3 + 4 \* 5 returned 23 but typing in (3 + 4) \* 5 returned 35 which shows that arithmetic in Haskell does work as expected when the correct syntax is used.
3. 17
4. “.gnirts a si sihT”

**Exercise 2:**

a), b), c) were done on terminal by following instructions

d) i) 42

ii) Int -> Int (integer -> integer)

iii) Int (integer)

1. You get an error when you try to evaluate double “three” as the double function doesn’t recognise the string “three” as an integer therefore it cannot perform the operation.
2. I tested the square function and it works i.e. when I type “square 2”, 4 is returned which is the correct answer to 2^2.

**Exercise 3:**

1. a b c = if a^2+b^2==c^2 then True else False
2. “isTriple 3 4 5” returned True which is expected, “isTriple 3 4 6” returned False which is expected.

**Exercise 4:**

1. leg1 x y = x^2-y^2

leg2 x y = 2\*x\*y

hyp x y = x^2+y^2

1. “leg1 5 4” returned 9 which was expected, “leg2 5 4” returned 40 which was expected, “isTriple 9 40 41” returned True which was expected.

**Exercise 5:**

1. The function prop\_triple works by using the functions leg1, leg2 and hyp to see whether the 2 inputs made buy the user would make a Pythagorean triple. The function expects an input of 2 integers that would be stored in the variables x and y, and in return, it outputs a Boolean answer (True/False).
2. “prop\_triple 3 4” returned True which was expected as 3 4 5 is a Pythagorean triple.

“prop\_triple 4 5” returned True

“prop\_triple 5 6” returned True

1. “quickCheck prop\_triple” returned “+++ OK, passed 100 tests.”