Reference: Big C++.

Exercises 4:

Exercise R4.1.

1. Sine or square root.

Exercise R4.2.

1. False.
2. True.
3. False.
4. True.
5. True.
6. False.
7. True.
8. True.

Exercise R4.3.

1. double sqrt (double x)

/\*\*

Computes the square root of a number

@param x the root to the square root function

@return the result of the operation

\*/

1. string roman\_numeral (int n)

/\*\*

Convert a numerical value to its corresponding in the Romanian number system

@param n is the numerical value to be converted, its type is integer, and this means that if the function was supplied with floating-point number, only the integer part will be converted

@return a string describes the roman numerical value

\*/

1. bool is\_leap\_year (year y)

/\*\*

Check whether a year is leap or not.

@param y is the year that would be checked.

@return a bool data type (true or false) to check whether a year is leap (true), or not (false)

\*/

1. string weekday (int w)

/\*\*

Gives the corresponding day name to the number given, for instance, if w = 7 then it returns Friday.

@param w is the number of the day and it must be less than 8 and greater than 0

@return is the name of the day

\*/

1. Point midpoint (Point a, Point b)

/\*\*

Computes the center of two points a, b

@param a is the first point

@param b is the second point

@return is a point describes the middle of the two points.

\*/

1. double area (Circle c)

/\*\*

Computes the area of a given circle.

@param c is the circle that would we compute its area

@return is the area computed

\*/

1. double slope (Line a)

/\*\*

Computes the slope of a given line

@param a is the line given to compute its slope

@return is the value of the slope

\*/

Exercise R4.4.

1. x1 = 39
2. x2 = 400
3. x3 = 92
4. x4 = 62
5. x5 = 0

Check code:

#include <iostream>

#include <cmath>

double f(double x);

double g(double x);

double h(double x);

double k(double x);

int main()

{

double x1 = f(2);

double x2 = g(h(2));

double x3 = k(g(2) + h(2));

double x4 = f(0) + f(1) + f(2);

double x5 = f(-1) + g(-1) + h(-1) + k(-1);

std::cout << x1 << " - " << x2 << " - " << x3 << " - " << x4 << " - "

<< x5 << "\n";

}

double f(double x) { return g(x) + sqrt(h(x)); }

double g(double x) { return 4 \* h(x); }

double h(double x) { return x \* x + k(x) - 1; }

double k(double x) { return 2 \* (x + 1);

Exercise R4.5.

A predicate function returns a Boolean value.

/\*\*

Tests whether the first number is greater than the other

@param x1 is the first number

@param x2 is the second number

@return true if x1 is greater than x2

\*/

bool is\_greater(double x1, double x2)

{

if (x1 > x2)

return true;

return false;

}

Exercise R4.6.

1. Parameter value is the value of the variable that has been given to a function as an input.

A return value is the value that will be returned as a result of the calculations of the function.

1. Parameter value is the value contained in the parameter variable ☺ .
2. Parameter value differs completely from value parameter.

Value parameter is a way of passing the function’s parameters

It means that take a copy of the variable and put it into the parameter variable,

and doesn’t mean take that variable and modify it like the reference parameter.

Exercise R4.8.

1. y = sin(x);
2. print\_paycheck(harry);
3. raise salary(harry, 5.5);
4. make uppercase(message);
5. key = uppercase(input);
6. change\_name(harry, "Horton");

Exercise R4.9.

The variables a and b are global which means that they are defined for all the functions below them.

The variables c and n are defined only in the function (f).

New variables c, n, and a are defined in the function (g).

Note that the variable a in the function g is not the same as the global variable a.

The variable i is defined only in the function main(), and a new variable b is defined only in main.

The program should print 3.

Exercise R4.10.

The variables a and b are global variables.

In the function f: the variable c is a parameter variable and the variable n is local to that function.

In the function g: the variable c is a parameter variable, the variable n and the variable a are local to that function.

In the function main: the variable i and the variable b are local to that function

::::

Note, to access the global variable b in main, you need to write (::b)

Read more, <https://goo.gl/tfgwL3>

Exercise R4.12.

void average(double& avg)

{

cout << "Please enter two numbers: ";

double x;

double y;

cin >> x >> y;

avg = (x + y) / 2;

}

This function has one parameter variable and doesn’t have return values.

Exercise R4.13.

* A function must have a return value, a procedure must not.

A procedure is a function that returns nothing and it usually has side effects.

* The function is a part of the program, several functions make up a program.
* The main function is the entry to the program; every program must have one function called main.

Exercise R4.15.

1. double sqrt(double x);

// precondition: x >= 0

1. double tan(double \_x)

// precondition: x == ±0.5

Exercise R4.17.

int f(int n)

{

if (n <= 1) return 1;

if (n % 2 == 0) // n is even

return f(n / 2);

else return f(3 \* n + 1);

}

This function will always return one for any number n.

Exercise R4.18.

Because after this statement a = b;   
the value of a has been now changed and lost forever.

The remedy is simple, save the value of (a) to a temporary variable.

Exercise R4.19.

Because the parameter variables were passed by reference.

This code works!

#include <iostream>

using namespace std;

void false\_swap2(int& a, int& b)

{

int temp = a;

a = b;

b = temp;

}

int main()

{

int x = 3;

int y = 4;

false\_swap2(x, y);

cout << x << " " << y << "\n";

return 0;

}

Exercise R4.19.

We can write the function tricky\_swap as follows:

void tricky\_swap(int& a, int& b)

{

int x = a - b;

b = x + b;

a = b - x;

}

If x = a – b

Then

b = x + b = a – b + b = a

And

a = b – x

And b now equals a

Then a = a – x = a – (a – b) = b

When the parameters refer to the same object, it returns zero!