|  |  |
| --- | --- |
| Edith Cowan University School of Science |  |

Workshop 7

Functions in C++

**Related Objectives:**

* Write simple functions
* Place functions in files
* Understand procedural abstraction
* Learn about scope rules
* Return values from and pass values to functions
* Learn to avoid common errors when using functions
* Use objects as arguments to functions and as return types of functions

**Activity:**

1. The output (to screen) of the following code snippet is: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**.

float compute(float a, float \*b, float &c)

{

a += 2;

(\*b)--;

c++;

return(a + \*b + c);

}

int main()

{

float x, y, z;

x = 8; y = 4; z = 0;

float w = compute(x, &y, z);

cout << x << "," << y << "," << z << "," << w << "\n";

return 0;

}

1. Consider the Vector3 struct which is defined in the code below:

struct Vector3

{

double x, y, z;

};

In the space provided below, write a function setVector which accepts four inputs: a reference to a Vector3 type variable, *beingSet*, and three individual double type numbers, *a*, *b* and *c*. The function should set the x, y and z components *beingSet* to *a*, *b* and *c* respectively.

When the setVector function is used as follows:

Vector3 oneVector;

setVector(oneVector, 1, 4, 5);

cout << oneVector.x << ", " << oneVector.y << ", " << oneVector.z << "\n";

The output should be: 1,4,5

1. Use the following code and perform the following tasks:
   1. Externalising both the read and write functions to their own external files
   2. Define your header file
   3. Change the menuSelect() function definition to be used as an inline function

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

string myRead(string filename);

int myWrite(string filename);

int menuSelect();

int main() {

string filename = “db.txt”;

int menuItem;

string line;

menuItem = menuSelect();

if (menuItem == 1) {

myWrite(filename);

system(“pause”);

}

else if (menuItem == 2) {

myRead(filename);

system(“pause”);

}

}

int menuSelect() {

int mItem;

cout << “This is a little test for C++ file handling. \n” << “to read from a file, type (1): \n” << “To write to a file, type (2): \n”;

cin >> mItem;

while(!(mItem == 1) && !(mItem == 2)) {

cout << “Wrong selection, do you want to type again? \nIf so, type (1) to read or (2) to write”;

cin >> mItem;

if(!mItem == 1 && !mItem == 2);

}

return mItem;

}

string myRead(string filename) {

long begin, end, lineNumber;

string record;

cout << “I’m in reading mode.\n”;

ifstream myfile(filename);

if(myfile.is\_open()) {

lineNumber = 0;

while(!myfile.eof()) {

getline(myfile, record);

cout << “Line Number: “ << lineNumber << “ – “ << record << endl;

++lineNumber;

};

myfile.close();

return record;

}

else cout << “Unable to open file”;

return 0;

}

int myWrite(string filename) {

cout << “I’m in writing mode.\n”;

ofstream myfile(filename, ios::app);

if(myfile.is\_open()) {

myfile << “You see I can write to your file.\n”;

myfile << “I can do it again.\n”;

myfile.close();

}

else cout << “Unable to open file”;

return 0;

}

Answers:

Q1: 8,3,1,14

Q2:

void setVector(Vector3 &beingSet, double a, double b, double c)

{

beingSet.x = a;

beingSet.y = b;

beingSet.z = c;

}

Q3: See solutions file on blackboard.