

# Exercise Class 9 – Theory

## 1 Classes and Operator Overloading

It is possible for two functions to have the same name, as long as the compiler has another way to differentiate between them. The possible criteria are number and type of the arguments of the function. The following is an example for a different number of arguments.

```
int fun1 (int a) { ... }
int fun1 (int a, int b) { ... }
```

And one for different types of arguments.

```
int fun2 (int a) { ... }
int fun2 (float a) { ... }
```

It is important to notice that the name of the variables is not a viable criterion.

```
int fun3 (int a) { ... }
int fun3 (int b) { ... } // compiler error
```

As isn't the return type.

```
int fun4 (int a) { ... }
double fun4 (int a) { ... } // compiler error
```

So now let's have a look at a more explicit example for overloaded functions:

```
#include <iostream>

void out (const int i) {
    std::cout << i << " (int)\n";
}

void out (const double i) {
    std::cout << i << " (double)\n";
}

int main () {
    out(3.5);           // 3.5 (double)
    out(2);             // 2 (int)
    out(2.0);          // 2 (double)
    out(0);             // 0 (int)
    out(0.0);          // 0 (double)
    return 0;
}
```