Mechatronic Engineering

Object Oriented Programming and Software Engineering
Laboratory instruction 5
C++ introduction

AGH Kraków, 2021

Materials created for educational purposes.
Dedicated for students attending Software Engineering course.
Author would appreciate any feedback regarding errors of any kind found in the instruction script.
Please report those to the following email address: danielt@agh.edu.pl

Table of contents

1 Enumeration 4

2 Typedef 5

3 Function overloading 6

1 Enumeration

Enumeration is user defined int type that consists internal constant and each integral constant is given a name (Such constant can be called identifier). To define enumerated integer data type, function enum is used.

```
enum type_name{value1, value2, ..., valueN};
```

type name is the name of enumerated data type. value1, value2,... are values of type name. By default, value1 will be equal to 0, value2 will be 1 and so on but, the programmer can change the default value as seen in the example (Sa=10).
Declaring enum variables:

```
enum type_name{value1, value2, ..., valueN} variable_name;
```

Example:

```
#include <iostream>
using namespace std;

int main(){
   enum week {Mo, Tu, We, Th, Fr, Sa=10, Su} WeekDay;
   WeekDay = Mo; cout << "Value for Monday = " << WeekDay <<endl;
   WeekDay = Tu; cout << "Value for Tuesday = " << WeekDay <<endl;
   WeekDay = We; cout << "Value for Wednesday = " << WeekDay <<endl;
   WeekDay = Th; cout << "Value for Thursday = " << WeekDay <<endl;
   WeekDay = Fr; cout << "Value for Fiday = " << WeekDay <<endl;
   WeekDay = Sa; cout << "Value for Saturday = " << WeekDay <<endl;
   WeekDay = Su; cout << "Value for Sunday = " << WeekDay <<endl;
   return 0;
}
```

```
#include <iostream>
using namespace std;
int main(){
   enum week{Mo, Tu, We, Th, Fr, Sa=10, Su};
   enum week today;
   today=We;
   cout << today+1 << " day" << endl;
   return 0;
}
```

2 Typedef

Typedef is used to define data type.

```
typedef type_definition new_type_name;
```

type definition is the data type that is going to be used in the new data type. The name of the new data type is set by the new type name. When new data type is created, it is possible to declare new variables of this type.

```
typedef float kg;
kg bear,tiger;
```

Typedef can be used to quickly change the variable types.

Example:

```
#include <iostream>
typedef float RealNumber;
typedef double OverloadedRealNumber;

int main(void){
   RealNumber a=4.5;
   RealNumber b=3566789.234576;
   OverloadedRealNumber c=698519801234.38254;
   OverloadedRealNumber d=3566789.234576;
   cout << "a = " << a << endl;
   cout << "b = " << b << endl;
   cout << "c = " << c << endl;
```

```
    cout << "d = " << d << endl;
    return 0;
}
```

3 Function overloading

overloading a function name means that there is more than one function with the same name in the given validity range. Activation of the function depends on the number and type of arguments to call it.

```
void function (int);

void function (int, char);

void function (float, int);
```

When reloading, the difference of arguments is important, the type returned by the function is not considered. The example bellow will generate an error.

```
void function (int);

float function (int);

void function (int, char);

void function (float, int);
```

In C++ it is possible to declare same element multiple times, the problem with functions starts during definitions. If the compiler finds multiple declaration of the same function (without overloading rule) it will ignore the duplicated declaration. But when compiler finds a second definition of an already defined function it will generate an error. There is however a way to overload a function

with function containing same argument types. It can be achieved by swapping the arguments in place:

```
void function (int, char);

void function (char, int);
```

One may ask what about enum, is it possible to overload a function with a type created with it? Unfortunately, the answer is no. Types created with enum are int from the fundamental types point of view.

Example:

```
#include <iostream>
using namespace std;

int area(int a){ return (a*a);}
int area(int a, int b){ return (a*b);}

int main() {
   int sideA = 2; int sideB = 4;
    cout << "There is a rectangle in which two squares are
       inscribed" << endl;
    cout << "The area of the square = " << area(sideA) << endl;
    cout << "The area of the rectangle = " << area(sideA, sideB)
    << endl;
return 0;
}
```

Task

Based on the information provided in previous manual, please add a folder called 'exercises', containing all example programs from all laboratory instructions including this one, to your remote repository.