ROBOT 4ALL

INTELLECTUAL OUTPUT 5

ROBOT4ALL
EXERCISES AND SOLUTIONS - CODING

Leading Partner: CCS

Co-funded by the
Erasmus+ Programme
of the European Union

CYPRUS
COMPUTER
SOCIETY

IMPRINT

PROJECT LEADERSHIP:

Prof. Dr. Dirk Lange

Leibniz Universität Hannover

Institut für Didaktik der Demokratie

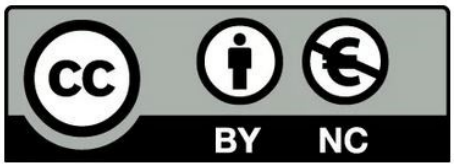
PROJECT MANAGEMENT:

Richard Heise

heise@idd.uni-hannover.de

LAYOUT:

Mareike Heldt



Co-funded by the
Erasmus+ Programme
of the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein. Project Number: 2017-1-DE02-KA202-004274

PROJECTS WEBSITE:

www.robovet.eu

CONSORTIUM

LEIBNIZ UNIVERSITÄT HANNOVER
Germany (Coordinator)

2 EK PEIRAIA
Greece

EMPHASYS CENTRE
Cyprus

CIVIC COMPUTING
United Kingdom

CDIMM MARAMURES FOUNDATION
Romania

CYPRUS COMPUTER SOCIETY
Cyprus

WOMEN IN DIGITAL INITIATIVES LUXEMBOURG ASBL
Luxemburg

I.E.S. MARÍA MOLINER
Spain

CONTENTS

INTRODUCTION

Purpose of the booklet

The present training material booklet complements the book with the teaching material of programming as it refers to the contents and the theoretical subjects of the specific chapters contained therein. It includes exercises on how students should approach the basic concepts and characteristics of programming as well as the techniques and methods used to teach the specific subjects.

This booklet is intended to be an accompanying aid to both student and teachers who teach the theory, to enhance the achievement of the teaching objectives of the theoretical part of the lesson.

The activities contained in the training booklet follow the framework of the project.

LESSON 1

Introduction to programming - Training activities (Practice)

- Visit and play the games in the following sites:
  https://www.allcancode.com/hourofcode
  https://codecombat.com/
- Write down and discuss your impressions.

Introduction to C++ - Training activities (Practice)

- Write down the pros and contras of another programming language you know
- Compare them with C++

Introduction to Pseudocode and Basic Programming Constructs - Training activities (Practice)

- Write down the pseudocode of Primary and secondary equation (ax+b=0 and ax2+bx+c=0).
- Write down the pseudocode of the basic low of Physics e=mc2. Can you produce pseudocodes of other lows of Physics?
- Three candidates participate in an election. Given the number of the votes, can you write down a pseudocode finding their percentages and who won the elections?
- Discuss your impressions

Flowcharts - Training activities (Practice)

- Create a flowchart for the following problem: A school would like to check whether its students have passed their courses. The passing grade is 10 out of 20. The system should notify the user accordingly depending on the results of each student. Also, the school would like to know the average mark of a student.
- Can you transform the pseudocode you created in the previous chapter to flowchart?

Starting our first project using Eclipse - Training activities (Practice)

- Download and set up your Eclipse installation.
- Discuss with your class if you experienced any problems .

Familiarizing with the Eclipse Editor - Editor Areas and Sections - Training activities (Practice)

- Following the steps described in the training material, create your first project.
- Are you familiar with a programming environment? Do you find any differences?

BASIC C++ - Training activities (Practice)

1. Write down a program that prints on the screen:

"Hello, world!
This my first program"

Solution:

```
#include <iostream>
#include <conio.h>
using namespace std;

int main ()
{
cout << "\nHello, world!";
cout << "\nThis is my first programa";
cout << "Press any key to continue...";
getch();
return 0;
}
```

```
Hello, world!
This is my first programa
Press any key to continue...
```

2. Write down a program that receives two inputs from keyboard. One of them should be a name to be stored in a string variable and the other should be an integer meaning a mark (from 0 to 10). The program prints an amusing message depending on the mark:

- if qualification < 5 prints then "Cheer up" with the name and "!Life is great.
- if qualification is >= 5 and < 7 then prints the name and "it is okey".
- if qualification is >=7 and <10 then prints the name and "you're really super".

Solution:

```
#include <iostream.h>

int main()
{
   char name[16];
   int qualification, tecla;

   cout << "Enter your name: ";
   cin >> name;
   cout << "Enter your qualification(0/10): ";
   cin >> qualification;

   if ( qualification < 5 )
      cout << "Cheer up " << name << "! Life is great\n";
   else if (qualification < 7)
      cout << name << ", it is okey!\n";
   else if (qualification < 10)
      cout << "You're really super, " << name << "!\n";
   else
      cout << "Are you the number one " << qualification << ", " << name << "?\n";

   cin >> tecla;
   return 0;
}
```

```
Enter your name: Enrique
Enter your qualification(0/10): 7
You're really super, Enrique!
```

3. Write down a program that performs a countdown from a number that you enter to 0 descending two by two, using a while-loop:

Solution:

```
// custom countdown using while
#include <iostream>
using namespace std;
#include <conio.h>
int main ()
{
int n;
char tecla;
cout << "Enter the number > ";
cin >> n;
while (n>0) {
cout << n << ", ";
n-=2;
}
cout << n << ".";
cout << "\nPress any key to continue";
getch();
return 0;
}
```

```
Enter the number > 20
20, 18, 16, 14, 12, 10, 8, 6, 4, 2, 0.
Press any key to continue
```

4. Write down a program that performs a countdown from a number that you enter to 0 descending two by two using a for-loop:

Solution:

```
// custom countdown using do-while loop
#include <iostream>
using namespace std;
#include <conio.h>
int main ()
{
int n;
cout << "Enter the number > ";
cin >> n;

do {
cout << n << ", ";
n-=2;
}while (n > 0);
cout << "0.";
cout << "\nPress any key to continue";
getch();
return 0;
}
```

```
Enter the number > 20
20, 18, 16, 14, 12, 10, 8, 6, 4, 2, 0.
Press any key to continue
```

5. Write down a program that accepts integer inputs until the user enters a zero to stop.

Solution:

```
// custom countdown using do-while loop
#include <iostream>
using namespace std;
#include <conio.h>
int main ()
{
int n;

while (true) {
cout << "Enter a number (0 to quit):> ";
cin >> n;
cout << "echo ... " << n << "\n";
if (n == 0) {
    cout << "loop aborted\n";
    break;
} }

cout << "\nPress any key to continue";
getch();
return 0;
}
```

```
Enter a number (0 to quit):> 98
echo ... 98
Enter a number (0 to quit):> -7
echo ... -7
Enter a number (0 to quit):> 12
echo ... 12
Enter a number (0 to quit):> -2
echo ... -2
Enter a number (0 to quit):> 0
echo ... 0
loop aborted

Press any key to continue
```

6. Write down a program that writes the sum of the pair numbers between 1 and 10. Note: the program must use 'the continue statement' and loop sentence 'for (i=2; i<=100; i++){}'.

Solution:

```
// custom countdown using do-while loop
#include <iostream>
using namespace std;
#include <conio.h>
int main ()
{
int i, sum=0;

for (i=2; i<=100; i++) {
    if (i%2 != 0) continue;
    // even number
    sum += i;

}
cout << "Sum: " << sum;
cout << "\nPress any key to continue";
getch();
return 0;
}
```

```
Sum: 2550
Press any key to continue
```

7. Write down a program that performs a countdown from a number that you enter to 0 descending two by two, using a goto-loop:

Solution:

```
// goto loop example
#include <conio.h>
#include <iostream>
using namespace std;
int main ()
{
int n;
cout << "Enter a number: ";
cin >> n;
loop:
cout << n << ", ";
n-=2;
if (n>0) goto loop;
cout << "0.";
cout << "\nPress any key to continue...";
getche();
return 0;
}
```

```
Enter a number: 20
20, 18, 16, 14, 12, 10, 8, 6, 4, 2, 0.
Press any key to continue...
```

8. Write down a program that accepts float numbers in every iteration of an infinite loop until the user enters 3.14. In every iteration, this program prints the number entered and its square. Note: If it is greater than 99.99 the program is broken with the exit sentence.

Solution:

```
Enter a float number: 2.8
Number: 2.8 square: 7.84
Enter a float number: 1.25
Number: 1.25 square: 1.5625
Enter a float number: 3.14

Ends the current program
Press any key to continue...
```

```
#include <conio.h>
#include <iostream>
#include <cstdlib>
using namespace std;
#define LIMIT 99.99f
int main ()
{
float n;
cout << "Enter a float number: ";
cin >> n;
bool stop = (n == 3.14f);
while (!stop) {
if (n > LIMIT) {
    cout << "\nExits the current program" << "\nPress any key to continue...";
    getche();
    exit(0);
}
cout << "Number: " << n << " square: " << n*n;
cout << "\nEnter a float number: "; cin >> n;
stop = (n == 3.14f);
}

cout << "\nEnds the current program" << "\nPress any key to continue...";
getche();
return 0;
}
```

9. Write down a program that receives an integer as an input and prints its cube only if it is equal to 1, 2 or 3. In other case, the program prints "The number is not 1, 2 nor 3".

Solution:

```
#include <conio.h>
#include <iostream>
#include <cstdlib>
using namespace std;
int main ()
{
int n;
cout << "Enter an integer number: ";
cin >> n;
switch (n) {
case 1:
case 2:
case 3:
cout << "Number: " << n << " Cube: " << n*n*n << "\n";
break;
default:
cout << "This number is not 1, 2 nor 3";
}
cout << "\nEnds the current program" << "\nPress any key to continue...";
getche();
return 0;
}
```

```
Enter an integer number: 3
Number: 3 Cube: 27

Ends the current program
Press any key to continue...
```

10. Functions

- Write down a function called cube which accepts one integer number as an input parameter and returns the cube of that value. Within of int main() {}, the function must be called with one arguments passed by value.
- Write down a function called welcome which accepts one name as input parameter and prints "Welcome " and the name. Within of int main() {}, the function must be called with one argument passed by value.

Solution:

```
#include <conio.h>
#include <iostream>
using namespace std;
int cube (int base) { return (base*base*base); }
int product (int a, int b)
{
return (a*b);
}

void wellcome (char name[])
{
cout << "\nWellcome " << name;
}

int main ()
{
char name [] = "Piedad\0";
wellcome (name);

cout << "\nThe cube of " << 8 << "is: " << cube(8);
cout << "\nPress any key to continue";
getche();
return 0;
}
```

```
Wellcome Piedad
The cube of 8is: 512
Press any key to continue
```

11. Arguments passed by reference: Write down a function called swap with two I/O parameters that swap their respective values. Within of an int main() {}, the function must be called with two arguments passed by reference.

Solution:

```
//Arguments passed by reference.
#include <conio.h>
#include <iostream>
using namespace std;
void swap (int& a, int& b)
{
int z;
z=b;
b=a;
a=z;
}
int main ()
{
int x=30, y=50;

cout << "\nPrevious: " << "x: "<< x << " y: " << y;
swap (x, y);
cout << "\nNow: " << "x: "<< x << " y: " << y;
cout << "\nPress any key to continue...";
getch();
return 0;
}
```

```
Previous: x: 30 y: 50
Now: x: 50 y: 30
Press any key to continue...
```

12. Parameter default values: Write down a function called divideBy2 with two input parameters (a,b) passed by value. This function returns the division of a/b. In case of not passing the second argument this function should assume "2"as second parameter giving a/2 as a result.

Solution:

```
//Arguments passed by reference.
#include <conio.h>
#include <iostream>
using namespace std;
double divideBy2 (int a, int b=2)
{
    return ((double)a/b);
}
int main ()
{
int x=30, y=50;

cout << "x: "<< x << " y: " << y;
cout << "  ,x/y: " << divideBy2(x,y);
cout << " , x/2: " << divideBy2(x);
cout << "\nPress any key to continue...";
getch();
return 0;
}
```

```
x: 30 y: 50  ,x/y: 0.6 , x/2: 15
Press any key to continue...
```

13. Working with the concept of overloaded functions: Write down a program that defines two functions with the same name to calculate the area of a square and the area of a triangle respectively.

Solution:

```
// overloaded function
#include <iostream>
#include <conio.h>
using namespace std;
float area (float l)
{
return ( l*l );
}
float area  (float b, float h)
{
return ( (b*h)/2 );
}
int main ()
{

float n=7.25,m=2.0;
cout << area (n);
cout << "\n";
cout << area (n,m);
cout << "\n";
cout << "Press any key to continue...";
getche();


return 0;
}
```

```
52.5625
7.25
Press any key to continue...
```

14. Write down a function called factorial that calculates the factorial of an integer number.

Solution:

```
// factorial calculator
#include <iostream>
#include <conio.h>
using namespace std;
long fact(long n)
{
if (n == 0) return 1;
else return (n * fact(n-1));

}
int main ()
{
long number;
cout << "Plense type n number: ";
cin >> number;
cout << number << "! = " << fact(number);

cout << "\nPress any key to continue...";
getche();

return 0;
}
```

```
Plense type n number: 9
9! = 362880
Press any key to continue...
```

15. Write down a program that includes the prototype of a function isOdd with a parameter passed by value. If it is odd returns true, in other cases, false.

Solution:

```
#include <iostream>
#include <conio.h>
using namespace std;
bool isOdd (int a);

int main ()
{
int n;
char msg_even [] ="This number is even\0";
char msg_odd [] ="This number is odd\0";

do {
cout << "\nType a number (0 to exit): ";
cin >> n;
if (!isOdd(n)) cout << msg_even; else cout << msg_odd;
} while (n!=0);

cout << "\nPress any key to continue...";
getch();
return 0;
}

bool isOdd (int a)
{
if ((a%2)!=0) return true;
return false;
}
```

```
Type a number (0 to exit): 8
This number is even
Type a number (0 to exit): 9
This number is odd
Type a number (0 to exit): 0
This number is even
Press any key to continue...
```

ADVANCED C++ - Training activities (Practice):

1. Working with the next concepts:

- Arrays
- Pointers
- Dynamic memory
- Data structures
- Pointers to structures
- Programming Oriented Objects

The computer enterprise "Golden computer" needs to register all its employees. The number of class to design should follow the next specifications:

Employee (Base class)

Properties

```
protected char * name;
protected char * surname;
protected char * id;
protectedint age;
protected long minSalary;
```

Methods

```
// constructor
Employee (char * name, char * surname, char* id, int age, long minSalary);
// destructor
~Employee(){
    cout<< "\nDestroying...";
    delete name; delete surname; delete id;}
```

// Write every method and interprets what is its functionality.

```
public char * getName();
public char * getSurname();
public char * getId();
publicintgetAge();
public long getMinSalary();
public virtual long calculateSalary()=0;
public virtual void visualiza() {};
```

Programmer is a class derived from the class Employee

Class Programmer (Derived class)

Properties

```
privateintanalysisByHour;
```

Methods

// Write the constructor

// Write the destructor

//Write the method that returns the number of lines written by the programmer in an hour.

intgetCodeLinesByHour();

//Write the mthod that returns the programming language's programmer.

char * getProgrammingLanguage();

// Write the method that returns the addition of minSalary and the money gained depending of the lines of code written in an hour. The hour costs 100 €.

longcalculateSalary();

Analyst is a class derived from the class Employee

Class Analyst (Derived class)

Properties

```
privateintanalysisByHour;
```

Methods

// Write the constructor

// Write the destructor

//Write the method that returns the number of hours worked by an analyst.

intgetAnalysisHours();

//Write the method that returns the programming language's programmer.

char * getProgrammingLanguage();

Once you have designed all the classes you'll realize a program -intmain() { return 0;} to test this exercise.

Steps:

- Declare an array that contains two positions occupied by references (pointers) to two objects instanced from the class Employee. Afterwards, prints the information of every position of that array. Remind, the value of every position is a reference of an object instanced from Employee.
- Test the destructor of Employee.

Solution:

```
#include <iostream>
#include <conio.h>
using namespace std;
class Employee {
        protected:
                char *name, *surname, *id;
                int age;
                longminSalary;
```

```
        public:
            Employee (char * name, char * surname, char* id, int age, long minSalary) {
                this->name = new char[30];
                this->name = name;
                this->surname = surname;
                this->id = id;
                this->age = age;
                this->minSalary = minSalary;
            };
            ~Employee(){
                cout<< "\nDestroying...";
                delete name; delete surname; delete id;}
            char * getName() { return this->name; }
            char * getSurname() { return this->surname ;};
            char * getId() { return this->id; };
            intgetAge() { return this->age; };
            longgetMinSalary() { return this->minSalary; };
            virtual long calculateSalary()=0;
            virtual void visualiza() {};
};
classProgrammer:public Employee {
        private:
            intcodeLinesByHour;
            char * programmingLanguage;
        public:
            Programmer(char * name, char * surname, char * id, int age, long minSalary,
            intcodeLinesByHour, char * programmingLanguage):Employee(
            name, surname,id , age, minSalary) {
                this->codeLinesByHour = codeLinesByHour;
                this->programmingLanguage = programmingLanguage;
            };
            intgetCodeLinesByHour() { return this->codeLinesByHour; };
            char * getProgrammingLanguage() { return this->programmingLanguage; };
            longcalculateSalary() { return minSalary + (this->codeLinesByHour *  10); };
```

```
voidvisualiza() {
                    cout<< "\n\n\n----------------------------------------------------";
cout<< "\nName: " << this->name << "\nSurname: " << this->surname
                    << "\nId: "<< this->id << " Age: " << this->age
                    << "\nMinimun salary: "<< this->minSalary
                    << "\nLanguage Programming: " << this->programmingLanguage
                    << "\nNumber of lines of code written: "<< this->codeLinesByHour
                    << "\n----------------------------------------------------\n";
          };
};

classAnalyst:public Employee {
          private:
                    intanalysisHours;
          public:
                    Analyst(char * name, char * surname, char * id, int age, long minSalary,
                    intanalysisHours):Employee(name, surname,id,age, minSalary) {
                              this->analysisHours = analysisHours;
                    };
                    intgetAnalysisHours() { return this->analysisHours; };
                    longcalculateSalary() { return minSalary + (this->analysisHours * 30); };
                    voidvisualiza() {
                              cout<< "\n\n\n----------------------------------------------";
cout<< "\nName: " << this->name << "\nSurname: " << this->surname
                              << "\nId: "<< this->id << " Age: " << this->age
                              << "\nMinimun salary: "<< this->minSalary
                              << "\nNumber of hours used in analysis: "<< this->analysisHours
                              << "\n----------------------------------------------------\n";
                    };
};
int main() {
          Employee *people[2];
          intopcion;
          int age;
```

```
                longminSalary;
                for (register inti=0; i<2; i++) {
                char * name = new char[30];
                char * surname = new char[50];
                char * id = new char[8];
                cout<< "\n\nEnter programmer o analyst: Programmer (1) - Analyst (2)" << "\n";
                cin>>opcion;
                cout<< "name?: (without white spaces) ";
                cin>> name;
                cout<< "surname?: (without white spaces) ";
                cin>> surname;
                cout<< "id?: (without white spaces) ";
                cin>> id;
                cout<< "age?: ";
                cin>> age;
                cout<< "min salary?: ";
                cin>>minSalary;
                if (opcion==1) {
                        intcodeLinesByHour;
                        char * programmingLanguage = new char[20];
                        cout<< "How lines of code you write by hour?: ";
                        cin>>codeLinesByHour;
                        cout<< "What's your programming language?: ";
                        cin>>programmingLanguage;
                        people[i]=new Programmer(name,surname,id,age,minSalary,
                        codeLinesByHour,programmingLanguage);
                }
                else {
                        intanalysisHours;
cout<< "How many hours do you use for the analysis of an application?: ";
                        cin>>analysisHours;
                        people[i]=new Analyst(name,surname,id,age,minSalary,analysisHours);

                }
}
```

```
// Prints people
for (register inti=0; i<2; i++) {
                    people[i]->visualiza();
                    cout<< "Total salary: " << people[i]->calculateSalary();
                    cout<< "\n\n";
}
delete people[0];
delete people[1];
cout<< "\n\nPress any key to continue...";
getche();
return 0;
}
```

Running:

```
Enter programmer o analyst: Programmer (1) - Analyst (2)
1
name?: (without white spaces) Soledad
surname?: (without white spaces) Ayuso
id?: (without white spaces) G-00001
age?: 45
min salary?: 123
How lines of code you write by hour?: 500
What's your programming language?: Java

Enter programmer o analyst: Programmer (1) - Analyst (2)
2
name?: (without white spaces) Pablo
surname?: (without white spaces) Galindo
id?: (without white spaces) G-00002
age?: 78
min salary?: 300
How many hours do you use for the analysis of an application?: 500



--------------------------------------------------
Name: Soledad
Surname: Ayuso
Id: G-00001 Age: 45
Minimun salary: 123
Language Programming: Java
Number of lines of code written: 500
--------------------------------------------------
Total salary: 50123




--------------------------------------------------
Name: Pablo
Surname: Galindo
Id: G-00002 Age: 78
Minimun salary: 300
Number of hours used in analysis: 500
--------------------------------------------------
Total salary: 150300

Destroying...
Destroying...

Press any key to continue...
```

2. Write down a template function that returns the smaller one of two objects. Use the function getMin within intmain() {return 0; } being the arguments:

- int
- long
- char
- double
- enumerated

Solution:

```
// function template
#include <iostream>
#include <conio.h>
using namespace std;
template<class myType>
myTypegetMin (myType a, myType b) {
myType result;
result = (a<b)? a : b;
return (result);
}

typedefenum { january=1, february, march, april,
may, june, july, august,
september, october, november, december} months_t;
int main () {
int result_1;
long result_2;
char result_3;
double result_4;
months_t result_5;

result_1=getMin<int>(5,8);
result_2=getMin<long>(123,34);
result_3=getMin<char>('a','z');
result_4=getMin<double>(125.6,23.99);
result_5=getMin<months_t>(january,december);

cout<< result_1 <<endl;
cout<< result_2 <<endl;
cout<< result_3 <<endl;
cout<< result_4 <<endl;
cout<< result_5 <<endl;
```

```
cout<< "Press any key to continue...";
getch();

return 0;
}
```

Running:

```
5
34
a
23.99
1
Press any key to continue...
```

3. Class Template:
   - Write down a class template called myTrio with three properties (a, b, c) of the type myTypeand the methods are:
   - A constructor (prototype) :myTrio(myType one, myType two, myType three){}
   - A method that returns the smaller one of these three properties (prototype): myTypegetMin() {}:
   - Think in a possible Running.

Solution:

```
// class templates
#include <iostream>
#include <conio.h>
using namespace std;
template<class myType>
classmyTrio {
myType a, b, c;
public:
myTrio(myType one, myType two, myType three){a = one; b = two; c = three;}
myTypegetMin() {
myType result;
if (a < b && a < c) result = a;
else if (b < a && b < c) result = b;
else result = c;
return result;
}
;
```

```
};

int main () {
myTrio<int> trio_1(100, 75, 23);
cout<< trio_1.getMin();
cout<< "\nPress any key to continue...";
getch();
return 0;
}
```

Running:

```
23
Press any key to continue...
```

4. Write down a program that declares one variable in a namespace called namespace1 and the same variable in another namespace called namespace2. Demonstrate that there are not redefinition errors:

Solution:

```
// namespaces
#include <iostream>
#include <conio.h>
using namespace std;
namespace namespace1
{
intmyVariable = 25;
}
namespace namespace2
{
double  myVariable= 80;
}
int main () {
cout<< namespace1::myVariable<<endl;
cout<< namespace2::myVariable<<endl;
cout<< "Press any key to continue...";
getch();
return 0;
}
```

Running:

```
25
80
Press any key to continue...
```

5. Write down a program that forces a bad_alloc exception and handles it printing several messages. You can try to allocate an array of 3000000000 ints;. This fact throws bad_alloc exception.

Solution:

```
// bad_alloc standard exception
#include <iostream>
#include <exception>
#include <conio.h>
using namespace std;

int main () {
        try   {
        int* myarray= new int[3000000000];
        } catch (bad_alloc& e)
        {
        cout<< "\nError allocating memory";
        cout<< "Standard exception: " <<e.what() <<endl;
        }

        cout<< "Press any key to continue...";
        getch();
        return 0;
}
```

Running:

```
Error allocating memoryStandard exception: std::bad_alloc
Press any key to continue...
```

6. Write down a program that converts the float number 315.398 into the type int.

Solution:

```
//Type casting
#include <iostream>
#include <exception>
#include <conio.h>
using namespace std;

int main () {
	float f=315.398;
	cout<< "Float number: " << f;
	cout<< "\nConverting a float number to int (Type Casting- Implicit conversion)";
	int c=f;
	cout<< "\nFloat number converted to int: " << c;
	cout<< "\nPress any key to continue...";
	getche();
	return 0;
}
```

Running:

```
Float number: 315.398
Converting a float number to int (Type Casting- Implicit conversion)
Float number converted to int: 315
Press any key to continue...
```

7. Write down a program that prints the real division of the expression 9/5 (explicit conversion – casting).

Solution:

```
//Type casting
#include <iostream>
#include <exception>
#include <conio.h>
using namespace std;

int main () {
	cout<< (float) 9/5;
```

```
        cout<< "\nPress any key to continue...";
        getch();
        return 0;
}
```

Running:

```
1.8

Press any key to continue...
```

8. Write down a macro that defines a function called getMin with two parameters passed by value. This function returns the smaller one of these two parameters. Test the macro.

Solution:

```
// function macro
#include <iostream>
#include <exception>
#include <conio.h>
using namespace std;

#define getMin(a,b) ((a)<(b)?(a):(b))

int main() {
int one, two;
cout<< "integer number 1?: ";
cin>> one;

cout<< "integer number 2?: ";
cin>> two;

cout<<getMin(one,two) <<endl;
cout<< "\nPress any key to continue...";
getch();
      return 0;
}
```

Running:

```
integer number 1?: 67
integer number 2?: 5
5

Press any key to continue...
```

9. Write down a program that writes in a text file the next phrase “Robots are amusing”.

Solution:

```
// basic file operations
#include <iostream>
#include <fstream>
#include <conio.h>
using namespace std;

int main () {
        ofstreammyfile;
        myfile.open ("robots.txt");
        myfile<< "Robots are amusing.\n";
        myfile.close();

        cout<< "Press any key to continue...";
        getch();

        return 0;

}
```

Running:

```
Press any key to continue...

A file called "robots. txt " has been created. Its first line contains:
“Robots are amusing”.
```