INM359 Object-Oriented Programming in C++
2020-21 Coursework – ***To be done individually***

Setting Up

First, make sure you have a ***proper*** environment to develop and test your code. You may use whichever IDE you want but your code ***MUST*** work with g++ version >=10.

So, **copy-paste** the following highlighted text in a **Mac**/**Linux**/**Cygwin** terminal:

```
mkdir tst; cd tst
cat > Makefile << EOF_Makefile
CXXFLAGS=-std=c++20 -pedantic -Wall -Wpointer-arith -Wwrite-strings
CXXFLAGS+=-Wcast-qual -Wcast-align -Wformat-security
CXXFLAGS+=-Wformat-nonliteral -Wmissing-format-attribute
CXXFLAGS+=-Winline -funsigned-char
CC=CXX
EOF_Makefile
cat > hello.cc << EOF_Hello
#include <iostream>
using namespace std;
#define X(token) #token
int main() {
  cout << X(Hello) << char(32) << X(World) << char(33) << endl;
  return 0;
}
EOF_Hello
make hello && ./hello
```

Does it print "Hello World!"? With no warnings/errors? Congratulations, you're good to go!
If the above doesn't work, contact me ASAP!!!
Debugging Tips:

- Are you using a **Windows** terminal? Stop it – use **Cygwin** (or **WSL+Ubuntu**).
- Are you using a shell that isn't Bash? Stop it – do "`echo $SHELL`" to see which shell you use and run "`bash`" to use the Bash shell in just this terminal.
- Did you try to ***type*** the above code yourself? Stop it – copy-paste it.
- Are you coding without enough sleep? Stop it – get some sleep first.
- `!!:gs/sleep/caffeine`

Your code should compile with no warnings using the Makefile file defined above (i.e., that set of compiler flags).

It's code files that you should submit, that can be compiled (q1.cc, q2.cc, …).

Questions

"Session 2 – Classes in C++"

1. Consider the following code. Fix the yellow parts, without modifying the parts in gray, so that the code compiles and runs correctly.

[10]

```
#include <iostream>
using namespace std;     // File: q1.cc
class some_class {
  int state;
 public:
```

```
  some_class(int s = 3) : state(s) {}

  int get_state();
  void set_state(int s);
};

/* Your code here */
                 //
                 //
                 //

int main() {
  some_class c1;
  c1.set_state(5);
  cout << c1.get_state() << endl; // should print 5
  return 0;
}
```

2. Consider the following use of class my_array_class. Fix the code in the class so that the grayed-out code compiles and runs correctly.
   Note: you should not modify any of the code in gray.

[20]

```
#include <iostream>
using namespace std;    // File: q2.cc
class my_array_class {
  int len = 2;
  int *a = new int [2];
 public:
  my_array_class() { a[0] = 1; a[1] = 2; }
  my_array_class(int ln, const int *o) : len(ln), a(new int [ln])
    { for (int n=0; n<ln; ++n) a[n] = o[n]; }
 ~my_array_class() { delete[] a; }

 int get(int n) {return a[n];}
 int set(int n, int v) { int tmp = a[n]; a[n] = v; return tmp; }
};

void foo( const my_array_class & a2 ) {
   std::cout << a2.get(0) << std::endl;
}

int main() {
  int zero12[] = {13, 1, 2};
  my_array_class a1(3, zero12);
  foo(a1);
  return 0;
}
```

"Session 3 – Overloading"

3. Extend the code in the class my_array_over below so that the grayed-out code compiles and runs correctly. (**Hint:** compare the grayed-out code here with Q2's.)

[20]

```
#include <iostream>
using namespace std;      // File: q3.cc
class my_array_over {
  int len = 1;
  int *a = new int [1];
 public:
  my_array_over() { a[0] = 0; }
  my_array_over(int ln, const int *o) : len(ln), a(new int [ln])
    { for (int n=0; n<ln; ++n) a[n] = o[n]; }
  ~my_array_over() { delete[] a; }
  /* Put your code here */
};
```

```
void foo( const my_array_over & a2 ) {
   std::cout << a2[0] << std::endl;
}
int main() {
  int zero12[] = {13, 1, 2};
  my_array_over a1(3, zero12);
  foo(a1);
 return 0;
}
```

4. Now do the same as in Q2 for the following code.

[20]

```
#include <iostream>
using namespace std;        // File: q4.cc
class my_array_over2 {
  int len = 1;
  int *a = new int [1];
 public:
  my_array_over2() { a[0] = 0; }
  my_array_over2(int ln, const int *o) : len(ln), a(new int [ln])
    { for (int n=0; n<ln; ++n) a[n] = o[n]; }
  ~my_array_over2() { delete[] a; }
/* Put your code here */
};

void foo( const my_array_over2 & a2 ) {
   std::cout << a2[0] << std::endl;
}

int main() {
  int zero12[] = {13, 1, 2};
  my_array_over2 a1(3, zero12);
  a1[1] = 14; /* EXTRA LINE!!! */
  foo(a1);
 return 0;
}
```

5. Consider the following class used for holding integer samples. Overload the output (**[20]**) and input (**[30]**) operators so that you can print and read objects of this class.

[50]

```
#include <iostream>
#include <vector>
using namespace std;        // File: q5.cc
class zample {
  vector<int> vi;
 public:
  zample(vector<int> some_vi) : vi(some_vi) {}
  const vector<int> & get_data() const { return vi; }
};

/* Put your code here */

int main() {
  vector<int> vi = {11, 12, 13, 14, 15};
  zample z1(vi);
  cout << z1 << endl; /* should print: <5: 11 12 13 14 15> I.e., <size: elements>
*/
  cin >> z1 >> z1; /* should be able to read "<5: 11 12 13 14 15><6: 21 22 23 24 25
26>a" (two samples on the same line, stuck together, with a character 'a'
immediately following the 2nd one. */
  cout << z1 << endl; /* should print: <6: 21 22 23 24 25 26> */
  char c;
  cin >> c; cout << c << endl; /* should print 'a' - the char after the
2nd zampleabove. */
  return 0;
```

```
}
```

"Session 4 – Genericity, Containers" & "Session 5 – Pointers and Arrays Iterators"

6. Consider the following code – complete it (in the yellow area) so that it compiles and runs.

[20]

```
#include <iostream>
#include <list>
#include <algorithm>
using namespace std;        // File: q6.cc

template <typename X>
class zamplex {
  list<X> vi;
 public:
  zamplex(list <X> some_vi);
  const list <X> & get_data() const;
};

/* a) Put your code here to get it to compile */

int main() {
  list<float> lf = {11, 12, 13};
  zamplex<float> z1(lf);
  const list<float> & lfr = z1.get_data();
/* b) Put your code here to print the elements of lfr. */
  return 0;
}
```

These questions consider how to implement the JOIN operator of SQL. It is based on the code in the appendix (which must be extended) and the data in the Wikipedia page for JOIN: https://en.wikipedia.org/wiki/Join_(SQL)
We have two tables – one describing employees and one describing departments and we need to compute their inner join, just as in the Wikipedia page.

7. Complete function print_dep so as to get a printout of the Department table. You should obtain something like:

```
DepartmentID DepartmentName
          31          Sales
          33    Engineering
          34       Clerical
          35      Marketing
```

[10]

8. Complete function print_emp so as to get a printout of the Employee table. You should obtain something like:

```
  LastName   DepartmentID
  Rafferty             31
     Jones             33
Heisenberg             33
  Robinson             34
     Smith             34
  Williams           NULL
```

[10]

9. Complete function print_inner_join so as to get a printout of the inner join of the Employee and Department table over their common DepartmentIDs. You should obtain something like:

```
  LastName   DepartmentID DepartmentName
  Rafferty             31          Sales
     Jones             33    Engineering
Heisenberg             33    Engineering
  Robinson             34       Clerical
     Smith             34       Clerical
```

[30]

10. Modify function print_dep so that it prints its results sorted by the ***department name***. Use the standard sort function – do **NOT** implement yours. You should obtain something like:

```
DepartmentID DepartmentName
          34       Clerical
          33    Engineering
          35      Marketing
          31          Sales
```

[10]

"Session 8 – Memory Management"

11. Here's a C++ program with a class called my_array_mem – it compiles fine with g++ >= 10, without any warnings but when it executes it crashes and does not print "1, 13, 3" as expected. Complete it so that it behaves correctly *(when compiled with g++ >= 10)*. The code inside functions main and foo below shows how my_array_mem is used.
Note: you should not modify any of the code in gray.

[20]

```
#include <iostream>
using namespace std;       // File: q11.cc
class my_array_mem {
  int len = 1;
  int *a = new int [1];
 public:
  my_array_mem() { a[0] = 0; }
  my_array_mem(int l, const int *o) : len(l), a(new int [l])
    { for (int n=0; n<l; ++n) a[n] = o[n]; }
 ~my_array_mem() { delete[] a; }
 int get(int n) {return a[n];}
 int set(int n, int v) { int tmp = a[n]; a[n] = v; return tmp; }
 /* Put your code here. Don't change the code in gray. */
};

void foo() {
  int zero12[] = {13, 14, 15};
  my_array_mem a1(3, zero12);
  my_array_mem a2 = a1;
  my_array_mem a3;
  a3 = a2;
  std::cout << a3.set(0, a2.get(1) ) << std::endl;
}

int main() {
 std::cout << 1 << std::endl;
 foo();
 std::cout << 3 << std::endl;
 return 0;
}
```

Attention to detail – Submitting

You need to submit ***code*** *(source code, not executables!)*, *not* a report in Word/PDF/etc.
So, create a folder named after your Moodle login (abcd123) and **copy** in there the files q1.cc, q2.cc, … along with the Makefile from page 1.
Zip your folder *(use zip, not rar/tar/etc.!)*.
Unzip your zip file **somewhere else**.
Try to compile all your files there – do they compile with the provided Makefile (hopefully with no errors/warnings)?
If so, submit your zip file (which should contain the Makefile and the source files only – no binary code!).

12%

Appendix

This appendix contains the code you need to expand for Q7-10.

Note: Since Word messes up the quotation marks (“”) used to denote strings in C++, we’ll use a trick of the C Pre-processor (cpp), which allows us to get a string out of a token (roughly what a valid name for a variable is – so no spaces allowed) – check out the macro “X”. In this way we ensure that there are no problems when copy-pasting the code from the coursework description.

```
#include <iostream>
#include <iomanip>      // std::setw
#include <sstream>      // std::ostringstream
#include <string>
#include <map>
#include <utility>      // std::pair
 /* add further includes here if you need to */
using namespace std;                  // File: q7.cc (for Q7–Q10)

/* Turn a token to a string - tokens cannot contain spaces or other characters that
are not valid for a variable name. */
#define X(token) #token

string null = X(NULL);

string department_names[] = {
   X(Sales)
   , X(Engineering)
   , X(Clerical)
   , X(Marketing)
};

string employee_names[] = {
   X(Rafferty)
   , X(Jones)
   , X(Heisenberg)
   , X(Robinson)
   , X(Smith)
   , X(Williams)
};

/* Table department: columns = DepartmentID, DepartmentName */
map<int, string *> department = { {31, department_names+0}
                                 ,{33, department_names+1}
                                 ,{34, department_names+2}
                                 ,{35, department_names+3} };
/* Table employee: columns = LastName, DepartmentID */
map<string *, int> employee = { {employee_names+0, 31}
                               ,{employee_names+1, 33}
                               ,{employee_names+2, 33}
                               ,{employee_names+3, 34}
                               ,{employee_names+4, 34}
                               ,{employee_names+5,  0} };

string get_null_or_X(int x) {
  ostringstream os;  os << setw(15);
  if (0 == x) os << null ;
  else        os << x ;
  return os.str();
}
string get_null_or_X(const string *x) {
  ostringstream os;  os << setw(15);
```

```
  if (nullptr == x) os << null ;
  else              os << *x ;
  return os.str();
}
string get_null_or_X(const char *x) {
  ostringstream os;
  os << setw(15) << x ;
  return os.str();
}

void print_dep() {
  cout << get_null_or_X(X(DepartmentID))
       << get_null_or_X(X(DepartmentName))<< endl;
  /* YOUR CODE HERE */
}

void print_emp() {
  cout << get_null_or_X(X(LastName))
       << get_null_or_X(X(DepartmentID))<< endl;
  /* YOUR CODE HERE */
}

void print_inner_join() {
  /* YOUR CODE HERE */
  cout << get_null_or_X(X(LastName))
       << get_null_or_X(X(DepartmentID))
       << get_null_or_X(X(DepartmentName))<< endl;
  /* YOUR CODE HERE */
}

int main() {
  print_dep();
  print_emp();
  print_inner_join();
  return 0;
}
```