Université d'Ottawa
Faculté de genie

École de science d'informatique
et de génie électrique

University of Ottawa
Faculty of Engineering

School of Electrical Engineering and
Computer Science

uOttawa

L'Université canadienne
Canada's university

CSI2372A
Advanced Programming Concepts with C++

FINAL EXAMINATION

Length of Examination: 3 hours **December 9, 2011, 9:30**

Professor: Jochen Lang **Page 1 of 16**

Family Name: _____

Other Names: _____

Student Number: _____

Signature _____

You are allowed **ONE TEXTBOOK** of your choice as a reference.
No calculators or other electronic devices are allowed.

Please answer the questions in this booklet.

If you do not understand a question, clearly state an assumption and proceed.

At the end of the exam, when time is up:

1) Stop working and turn your exam upside down.
2) Remain silent.
3) Do not move or speak until <u>all</u> exams have been picked up, and a TA or the Professor gives the go-ahead to leave.

	Marks	Out of
A		11
B.1		3
B.2		2
B.3		3
B.4		3
B.5		2
C		16
Total		40

PART A: Short Questions (11 POINTS)

For all questions in this part containing code excerpts assume that the necessary standard headers and using statements have been included.

1. Use a reverse iterator to loop and print (to `cout`) all the elements of type `string` stored in a `std::vector`

2. Given the function `isZero` use a `typedef` to name a pointer to such a function `zeroComp` and make the following program work correctly.

```
bool isZero( int );
// Give typedef



int main() {
  zeroComp ptrIsZero = isZero;
  ptrIsZero(1);
  return 0;
}
```

3. Use the function `std::replace_if` to replace all numbers equals 0 in the vector `std::vector<int> ivec;` with the number 1.

```
template <class ForwardIterator, class Predicate, class T>
void replace_if(ForwardIterator first, ForwardIterator last,
                Predicate pred, const T& new_value)
```

`Replace_if` replaces every element in the range `[first, last)` for which `pred` returns `true` with `new_value`.

4. The following code does not work as intended. Explain the error.

```
class A; class B;
A *objA = new A();
try {
  B* objB= dynamic_cast<B *>(objA);
} catch ( std::bad_cast& except ) {
  cout << "Cast failed!" << endl;
}
```

5. Use the function `std::copy`, to copy the content of the array `int tab[5]= {1,4,2,7,8};` into `std::vector<int> v`;

6. What does the following program print?

```
std::auto_ptr<int> pi(new int(4));
std::auto_ptr<int> pj = pi;
cout << pi.get() << endl;
```

7. Modify the following such that it throws an exception of type `char` with a value `'N'` when the argument `x` is negative.

```
int fct(int x) {


    x+= 2;
    return 3*x;
}
```

8. Given a class `Point` with class variables `d_x` and `d_y` . O, define the necessary operation to convert a `Point` to an integer (the result should be `d_x*d_x+d_y*d_y`).

9. Allocate dynamic memory for a two-dimensional array with 5 columns and 3 rows and assign it to the pointer `int **tab;`

10. Correct the error(s) in the following function by changing the function definition

```
int& func(const int& a) // Only change this line
{
   a++;
      return a;
}
```

11. Declare the variable `ret` such that it works correctly in the following code snippet.

```
std::map<int,float> m;
ret= m.insert(std::make_pair<int,float>(3,4.5));
```

PART B: Questions on Selected Topics (13 Marks)

1. Implement the assignment operator for class A such that it implements a deep copy strategy [3].

```
class A {
  int* d_a;
  int d_sz;
 public:
  A(int sz=0) : d_sz(sz) {
    d_a = new int[d_sz];
  }
  // other stuff
};


int main() {
  A a1(3), a2;
  a2=a1;
  a1=a1;
  return 0;
}
```

2. Give an implementation for the postfix increment operator for class Integer [2].

```
class Integer {
  int i;
public:
  // other stuff
};

int main() {
  Integer myInt;
  myInt++;
  return 0;
}
```

3. Make class A a template with the array type and array size as template parameters and change the main function to behave as before [3].

```
class A {
  static const int sz=3;
  double array[sz];
};

int main A {
   A a;
};
```

4. Add the functionality to the `class Couple` for the code below to give the desired output [3].

```
#include <iostream>

class Couple {
    int a,b;
 public:
   Couple(int a, int b) : a(a), b(b) {}
   // …
};

int main() {
  Couple  c(3,2);
  // Desired output:  c=(3,2)
  std::cout << “c=” << c;
  return 0;
}
```

5. Define a function that creates a file and stores the numbers `1 2 3` in a single line. The file name should be passed as argument to the function. [2]

PART C: Programming Questions (16 Marks)

The class `BookShelf` is used to organize a bookcase of N shelves. The books on each shelf are entered in a `std::list`. For greater flexibility, the class `BookShelf` is defined using a template.

```
#include <iostream>

template <typename T, int N=3>
class BookShelf {

  std::list<T> shelves[N];

public:

  T operator()(int shelfNumber, int elementNumber) const;
  BookShelf<T,N> operator+=(const T& element);
  BookShelf<T,N> operator--();
  int getNumberOfShelves() const {return N;}
  bool find(const T &element) const;
};
```

1. Suppose a class `Livre` has been defined, declare a `bookcase` (an object of type `BookShelf`) which contains 5 shelves [2].

2. Define the operator += such that it inserts an element in the shelf with the fewest elements (in case two shelves have an equal number of elements, use the shelf with the lower index) [3].

3. Define the operator `()` such that the call `bookcase(i,j)` returns the j^{th} element in the shelf `i` by value [3].

4. Define the operator -- such that `--bookcase;` removes the last element of the shelf that contains the most elements. [3].

5. Define the method `find` returning `true` if the element is in the `BookShelf`. Your function must use `std::find` for each shelf in the class `BookShelf` [3].

6. The class `BookShelf` does not contain a default constructor, copy constructor, assignment operator = or a destructor. Explain why those are not necessary for this class [2].