Université d'Ottawa
Faculté de génie

École de science
d'informatique
et de génie électrique

u Ottawa

L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical
Engineering
and Computer Science

CSI2372A
Advanced Programming Concepts with C++

MIDTERM EXAMINATION

Length of Examination: 75 minutes

October 10, 2013, 16:00

Professor: Jochen Lang

Page 1 of 10

Family Name: _____

Other Names: _____

Student Number: _____

Signature _____

You are allowed **ONE TEXTBOOK** 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: Stop working and turn your exam upside down.
Remain silent.

Question	Marks	
1.1-1.6		6
2.1		3
2.2		3
2.3		1
4a		5
4b		3
4c		3
4d		2
Total		26

PART 1: SHORT QUESTIONS (6 MARKS)

1. What is the value of `a` at the end of the following program

   ```
   unsigned int a = 0xFF, b= 0x0F0F;
   a ^= b;
   ```

   0xFF0

2. Given the following declarations
   ```
   class A {};
   class B : protected A {};
   A foo( A a );
   ```

   Find the error below and state why it is an error:
   ```
   B b;
   A* a = new A();
   foo(*a);
   A c = foo(b); // [Error] 'A' is an inaccessible base of 'B'
   *a = foo(c);
   ```


3. What is printed by the following program?

   ```
   union Combi {
     int i;
     short s;
   };
   ...
   Combi c;
   c.i = 0xFF0001;
   std::cout << c.s;
   ```

   1

4. What is the value of `a` at the end of this program?

```
int A[][2] = {10,20,30,40};
int (*ptr)[2] = &A[0];
int a = **(++ptr);
```

30

5. Given the function declarations below which one will be called by the code (or will the call be in error)?

```
void func(int& i);
int func(const float* f);  // Called, no error
float func(double* d);

float f=5.0f;
float g = func(&f);
```

6. Given:

```
class B {};
struct A {
    A( B b );
};
```

Call the function `void test( A a, int* b=0);` with the two corresponding variables below.

B b; int i;

test(b, &i);

PART 2: PROGRAMMING QUESTIONS (20 MARKS)

1. Bit manipulation: Give a function that changes the order of the bytes of a 32 bit (4 byte) unsigned integer. (Example: It should change 0xA50F8421 into 0x21840FA5) [3].

```
unsigned int changeByteOrder( unsigned int val32 );
```

```
unsigned int changeByteOrder( unsigned int val32 ) {
  return ((val32 & 0xFF) << 24) | ((val32 & 0xFF00) << 8) |
    ((val32 & 0xFF0000) >> 8) | ((val32 & 0xFF000000) >> 24);
}
```

2. Implement the following function to convert the ASCII characters in an old style C-string from lower case letters to upper case letters. (Characters which are not lower case should be unchanged). ASCII characters with decimal values 65-90 correspond to A-Z while characters with decimal values 97-122 correspond to a-z [3].

```
void makeUpper( const unsigned char* input, unsigned char* output);
```

```
// assumes output is large enough!

void makeUpper( const unsigned char* input, unsigned char* output ) {
  while ( *input != 0 ) {
    *output = (*input>=97 && *input<=122)?*input-32:*input;
    ++output; ++input;
  }
  *output = *input;
  return;
}
```

3. State what happens if your routine from the above question is called as follows [1]:

```
  unsigned char in[] = "university";
  unsigned char* res;
  makeUpper(in,res);
```
Illegal memory access because res is not pointing to valid memory.

4. Consider the below definitions for Room and Event. (Partial class definitions to be extended by you in the following questions).

```
#include <string>

struct Room {
  int d_noSeat;
  bool d_hasProjector;
};

class Event {
  Room* d_room;
  std::string d_name;
public:
  Event() : d_room(0), d_name("") {};
  void set( Room r, const std::string& name );
};
```

a. Implement any missing functions and constructors for Room and Event for the code below. Do you need to implement a constructor for room? If yes implement it, if no state why? [5]

```
int main() {
  Room r;
  r.d_noSeat = 37; r.d_hasProjector = true;
  Event lectures[25];

  for ( int i=0; i<25; ++i ) {
    lectures[i].set( r, "CSI2372" );
  }
  return 0;
}
```

```
void Event::set( Room r, const std::string& name ) {
  if ( d_room == 0 ) d_room = new Room(r);
  else *d_room = r;
  d_name = name;
  return;
}
```

b. Implement a shallow copy constructor for `Room`. Your implementation should behave exactly as the synthesized copy constructor by the compiler. Also modify the structure definition.[3]

```
struct Room {
  int d_noSeat;
  bool d_hasProjector;

  Room( const Room& _oRoom );



};

Room::Room( const Room& _oRoom ) :
                d_noSeat( _oRoom.d_noSeat ),
                d_hasProjector( _oRoom.d_hasProjector )
  {}
```

c. Implement a deep copy constructor for the class `Event`. Also modify the class
    definition.[3]

```cpp
class Event {
  Room* d_room;
  std::string d_name;
public:
  Event() : d_room(0), d_name("") {};


  Event( const Event& _oEvent );



  void set( Room r, const std::string& name );
};



  Event::Event( const Event& _oEvent ) : d_room(0),
    d_name(_oEvent.d_name) {
    d_room = new Room(*_oEvent.d_room);
  }
```

d. Implement a destructor for the class `Event`. Also modify the class definition.[2]

```cpp
class Event {
  Room* d_room;
  std::string d_name;
public:
  Event() : d_room(0), d_name("") {};

  ~Event();



  void set( Room r, const std::string& name );
};


Event::~Event() {
    if ( d_room != 0 ) delete d_room;
}
```