

Université d'Ottawa  
Faculté de génie

École de science  
d'informatique  
et de génie électrique



uOttawa

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 5, 2016, 14:30**

**Professor: Jochen Lang**

**Page 1 of 11**

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	Maximum
A.1-A.9		9
B.1		2
B.2		2
B.3		3
C.1		3
C.2		4
C.3		3
Total		26

## **PART A: SHORT QUESTIONS (9 MARKS)**

1. Given the following declaration

```
void arrays( int (*a)[3], int (&b)[3] );
```

call the function `arrays` with the arguments

```
int argA[3], argB[3];
```

2. What is printed by the following?

```
int array[][2]{1,2,3,4,5,6};  
cout << (*(array+1))[2] << endl;
```

3. What is printed by the following?

```
bool bv = true;  
short sv = 2;  
int iv = 1;  
cout << (iv < sv && bv) << endl;
```

4. What is printed by the following?

```
unsigned int ua = 4, ub = 2;
cout << (ua ^ ub | 1) << endl;
```

5. What is printed by the following?

```
char cA[]{"Hello World"};
*(cA+5) = 0;
cout << cA << endl;
```

6. What is printed by the following?

```
char abc[]{"abc"};
for ( auto v : abc ) {
    v++;
}
cout << abc << endl;
for ( auto& v : abc ) {
    v++;
}
abc[3] = 0;
cout << abc << endl;
```

7. What is printed by the following?

```
int aA[][2]{1,2,3,4,5,6};
int (*ptrA)[2] = aA;
++ptrA;
cout << (*ptrA)[1] << endl;
```

8. What is printed by the following?

```
int i=7, j=2;
auto k = i/j;
auto m = i%j;
cout << k << " and " << m << endl;
```

9. What is printed by the following?

```
int i = 2;
int& j = i;
auto k = j;
decltype(j) m = j;
--i;
cout << k << endl;
cout << m << endl;
```

## **PART B: Short Programs (7 MARKS)**

1. Mark the illegal access to union variables and indicate which variable was not supposed to be accessed [2]

```
union ShortLong {
    long lVal;
    short sVal;
};

int main() {
    ShortLong SL;
    SL.lVal = 1024L;
    cout << SL.sVal << endl;
    SL.lVal *= SL.sVal;
    SL.sVal = static_cast<short>(1024);
    SL.sVal = ++SL.lVal;
    return;
}
```

2. Complete the function `printEnum` such that it prints the foreground color, e.g., "Black on Black" or "White on Black". [2]

```
enum class Colors {  
    White, Black  
};  
  
int main() {  
    Colors foreground = Colors::White;  
    printEnum( foreground );  
    return 0;  
}
```

```
void printEnum( Colors foreground ) {
```

```
    cout << " on Black" << endl;  
    return;  
}
```

3. Complete the function `checkPairs` below to test if a hand of five cards contains a pair (i.e., 2 cards of the same face). If yes return true, otherwise false. [3]

```
enum class Color { Spades, Clubs, Hearts, Diamonds } ;
enum class Face { Seven, Eight, Nine, Ten, Jack, Queen, King, Ace };

struct Card {
    Color color;
    Face face;
};

bool checkPair(Card (&hand)[5]) {
```

## **PART C: PROGRAMMING QUESTIONS (10 MARKS)**

1. Consider the following definitions of the class `GameScore` with its helper structure `Game`.

```
/**
 * stores a player and his/her score
 */
struct Game {
    string player;
    int score;
};

class GameScore {
    // Array of games stored in GameScore
    Game d_games[10];
    // Counter of games added to GameScore
    int d_numGamesStored=0;

public:
    // Add a new Game to the array of games, return true on success.
    // If the array is full return false and do not store the Game.
    bool addGame( const Game& p );

    // Return the Game with the highest score
    // If no Game is stored, return a nullpointer
    Game* getHighScore();

    // Print all the games stored in d_games
    void print();
};
```



- a. Implement `GameScore::addGame` to add a new `Game` to the array of games and return `true`. If the array is full return `false` and do not store the `Game`. [3]

```
bool GameScore::addGame( const Game& p ) {
```

- b. Implement `GameScore::getHighScore` to return the game with the highest score. If no `Game` is stored, return a nullpointer [4]

```
Game* GameScore::getHighScore() {
```

- c. Implement `GameScore::print` to print all the games stored in `d_games` to console. [3]

```
void GameScore::print() {
```