Imed Hammouda

OHJ-1156 Programming II

Class Exam

Monday 16 March 2009

Make sure you read the questions carefully before giving your answer. Put your name and student number on each answer sheet.
This exam consists of **3** pages / **7** exercises. The maximum amount of points is 70. Each exercise is worth 10 points.
Written material, mobile phones and calculators are NOT allowed in the exam.
Good luck!

Exercise 1

(a) Write a recursive function definition for the following function:

```
int squares (int n);
// Precondition: n>= 1
// Returns the sum of the squares of numbers 1 through n.
```

For example, squares(3) returns 14 because $1^2 + 2^2 + 3^2$ is 14.

(b) Briefly explain the different types of recursion studied in the class. Give example C++ code for each type.

Exercise 2

(a) What is the output produced by the following code?

```
int *p1, *p2;
p1 = new int;
p2 = new int;
*p1 = 10;
*p2 = 20;
cout << *p1 << " " << *p2 << endl;
p1 = p2;
cout << *p1 << " " << *p2 << endl;
*p1 = 30;
cout << *p1 << " " << *p2 << endl;
```

How would the output change if you were to replace

```
p1 = p2;
```

with the following

```
*p1 = *p2;
```

(b) Briefly explain the relationship between *pointer* and *array* types in C++!

(c) *Tutnew* library reports the following error situations:
- Memory leaks (memory not released at the end of program)
- The same memory released several times
- Attempts to release memory which has not been allocated
- Array memory block released as normal memory or visa versa
- Released memory changed after release

For every situation above, give example C++ code that results in the error.

Exercise 3

Given the declarations

```
struct  NodeType;
typedef  NodeType* NodePtr;
struct NodeType
{
        int  key;
        NodePtr  link;
}
```

(a) Write a C++ function *int count(NodePtr, int)* to count the number of elements whose key value is equal to an input value.

(b) Write a C++ function *NodePtr removeFirst(NodePtr)* that deletes the first element of a linked list and returns a pointer to the first element of the new list. Handle properly the case of empty list!

(b) Write a similar C++ function that deletes the last element of a linked list.

Exercise 4

True or False?

(a) A function template should be split across separate declaration and implementation files.

(b) Instantiation is the process of building individual functions/classes by specifying actual parameters to the template.

(c) C++ restricts the number of instances from one template.

(d) In a definition of a function template, template parameters may or may not be used.

(e) In template declaration, the two keywords typename and class can be used interchangeably.

Exercise 5

(a) Given the following declaration for the class *Human*

```
class Human {
public:
  Human(string name, string address, int age);

  string getName() const;
  string getAddress() const;
  int getAge() const;

  void setName(string name);
  void setAddress(string address);
  void setAge(int age);

private:
  string name;
  string address;
  int age;
};
```

Based on class Human, write the declaration for a class *Family*. A Family object should contain at least three private data members: a father, a mother and list of children. The class Family should have one constructor that takes a father and a mother as parameters.

You should include public accessors and mutators to set and get the data members. In addition, you should provide a public method *AddChild*, that adds a child to the list of children. The maximum number of children is 20.
You only need to write the class declaration (i.e. the .h file), NOT the implementation of these functions. Your class declaration will be graded based on style and correctness.

(b) Write the implementation for the class Family member function: *getNumberOfChildren(int age)* that return the number of family children older than or equal to an input age value. (You do not need to add this function to your declaration of the Family class above.)

Exercise 6

(a) Give two strong reasons why *separate compilation* (compiling each file separately) is important. What tool is used for the purpose?

(b) Explain shortly parameter passing in the C++ *main* function.

(c) Briefly, explain the difference between *white-box testing* and *black-box testing*. List one example of white-box testing.

(d) Briefly, explain and indicate the objectives of *RCS*.

(e) Briefly, explain and indicate the objectives of *GDB*.

Exercise 7

(a) Briefly explain the three mechanisms how to use *name spaces*. Give code example for each.
(b) Briefly explain how *exception handling* should be organized among functions. Give code example.