

January 25, 2005



CMPS 258 Final Exam



American University of Beirut

الجامعة الأمريكية في بيروت



CMPS 258 Final Exam

January, 25, 2005

Student Name: _____ Student ID: _____

Signature: _____ Section: _____

There are fourteen pages, including this one. The test is out of 100 marks, and the value of each question is provided. Please use this information to manage your time effectively.

Question 1: _____/10
Question 2: _____/05
Question 3: _____/15
Question 4: _____/05
Question 5: _____/10
Question 6: _____/05
Question 7: _____/05
Question 8: _____/05
Question 9: _____/05
Question 10: _____/05
Question 11: _____/10
Question 12: _____/10
Question 13: _____/10
Total: _____/100

Question 1: imperative and Others [10 marks]

A) What is the difference between the **Von Neuman** machine and the logical declarative languages, when it comes to assignment statements? Explain. [2 marks].

B) The four main language design criteria are: Readability; Writability, Reliability; and Cost. Based on the languages we have studied, give a specific example where two criteria work against each other, and two examples of languages which chose to favor opposite criteria. Use a short code example if it will help support your answer. (In other words, say "In Language X and Language Y, the "foo" construct creates tension between Criterion A and Criterion B. Language X chose to favor A at the detriment of B, while Y favors B over A. Here's some code demonstrating this.") There are many examples you could cite here--pick a strong example. [6 marks]

C) What are the six main attributes of variables? [2 marks]

Question 2: Grammar and parsing [5 marks]

A) Here is a somewhat peculiar expression grammar:

$\langle \text{exp} \rangle \rightarrow \langle \text{factor} \rangle * \langle \text{exp} \rangle \mid \langle \text{factor} \rangle / \langle \text{exp} \rangle \mid \langle \text{factor} \rangle$

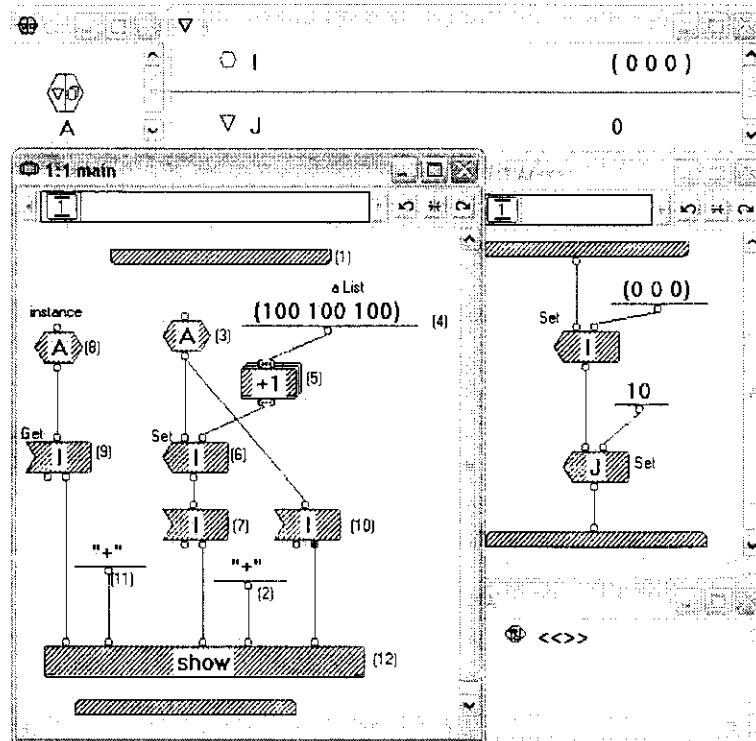
$\langle \text{factor} \rangle \rightarrow \langle \text{var} \rangle + \langle \text{factor} \rangle \mid \langle \text{var} \rangle - \langle \text{factor} \rangle \mid \langle \text{var} \rangle$

$\langle \text{var} \rangle \rightarrow A \mid B \mid C \mid D$

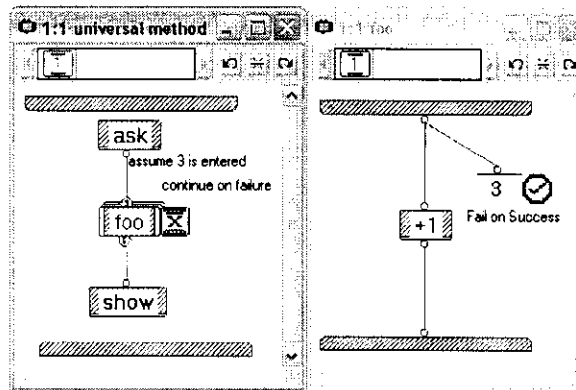
Show a parse tree for the expression: $A * B - C - D$

Question 3: Program Execution [15 marks]

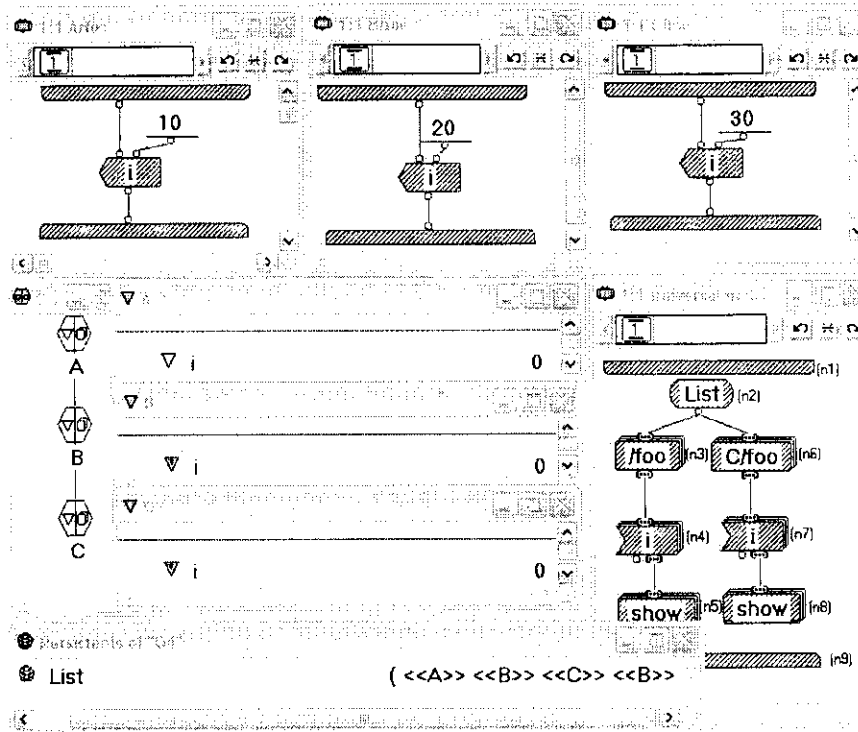
A) The following prograph code depicts a number to the right of each operation in function "main". Assuming that the execution order starts at (n1), and sequentially ends at (n12), what is the output of the following program? [5 marks]



B) What is the output of the following visual code? [5 marks].



C) What is the output of the following visual code? [5 marks]



Question 4: Features of Visual Dataflow Languages [5 marks]

A) Discuss the possibility of creating a dataflow language that is textual rather than visual. Describe how you would go about doing this. [4 marks]

B) It is said that visual dataflow languages help programmers create a mental image of the program execution thread. True or False? [1 mark]

Question 5: Inheritance, Substitutability and Polymorphism [10 marks]

A) Describe briefly two problems with multiple inheritance in C++. What is the role of virtual inheritance in C++, and what problem does it solve. Give a C++ example to demonstrate. Write code that makes sense. [4 marks]

B) Advanced features in C++ allow the following declarations in structures. This code does not compile. What would be the problem? [3 marks] **Note** that you need to discuss the concept behind lines 4, 5, and 6 in details for more marks. For each problem in the code suggest a solution.

```
#include <iostream>
using namespace std;
struct S {
```

```
virtual void f1() {cout << "f1() in S!" << endl;}  
class C: private S {  
    void f1()          { cout << "f1() in C!" << endl;}  
int i;  
};  
1- int main()  
2- {  
3- C c;  
4- S *sp = &c;  
5- sp->f1();  
6- sp->i = 10;  
7- return 0;  
}
```

C) Give a C++ example to illustrate the concept of (Class instance Record) (**CIR**) and their virtual tables or **Vtables**. Draw the necessary graphical representation and the C++ code to illustrate the concept of substitutability, and it's effect on the Virtual table pointer. **[3 marks]**

Question 6: General OO [5 marks]

A) Discuss briefly the advice that "all objects initialized by a constructor must be explicitly deleted somewhere". [3 marks]

B) What is the difference between a struct and a class in C++? [2 marks]

Question 7: Template metaprogramming [5 marks]

In the following code, the compiler attempts to instantiate `Factorial<6>`, it finds it must also instantiate `Factorial<5>`, which requires `Factorial<4>`, and so on. Eventually the recursion ends with the specialization `Factorial<1>`, and the computation unwinds. Eventually, `Factorial<6>::val` is replaced by the integral constant 720, and compilation ends. Since all the computation is done by the compiler, the values involved must be compile-time constants, hence the use of `enum`. When the program runs, the only work left to do is print that constant followed by a new line. Discuss the advantages and disadvantages of such programming approach.

```
#include <iostream>
using namespace std;
template <int n> struct Factorial {
    enum { val = Factorial<n-1>::val * n };
};
template <> struct Factorial<0> {
    enum { val = 1 };
};
int main() {
    cout << Factorial<6>::val << endl; // 720
    return 0;
}
```

Question 8: Private Data Members Scope [5 marks]

What is the output of the following code? And how was the concept of private data violated?

```
class A {
  int X;
  static int Y;
public:
  A(int x): X(x) {}
  int& foo()      {return X;}
  int& bar()      {return Y;}
  void foobar(A x){cout << x.X << endl; cout << x.Y << endl;}
};
//-----
int A::Y = 10;
int main(void)
{
  A a(5);
  int &i = a.foo();
  i++;
  a.foobar(a);
  int &j = a.bar();
  j++;
  a.foobar(a);
  return 0;
}
```

Question 9: Inheritance in C++ [5 marks]

what is the output of the following code?. What would happen and why, when we replace the line 20 with the line number 21. i.e remove line 20 and replace it with line 21.

```
#include <iostream>
using namespace std; //introduces namespace std
class Base {
public:
    Base(){ pub = 1; pro = 2; cout << pub <<" " << pro << endl; }
    int pub;
protected:
    int pro;
private:
    int pri;
};
class Derived: private Base{
public:
    Derived() {pub=10; pro=20; cout<<pub<<" " << pro << endl; }
};
class Derived_1: private Derived {
public:
    Derived_1() {pub = 10; cout << pub << endl; } // line 20
    // Derived_1() {} replaces the constructor. // line 21
};

int main( void )
{
    Base b;
    Derived d;
    Derived_1 d_1;
    return 0;
}
```

Question 10: Templates and Type Checking [5 marks]

What would be the output of this C++ Program? What would be the problem if the line that is commented is un-commented? Explain, and cite the advantages of templates.

```
#include <iostream>
using namespace std;
template <class Type>
class A {
public:
    Type CDataMember, result;
    public:
    A(Type x) : CDataMember(x) {}
    Type Get();
    void Set(Type x);
};
//-----
template <class Type>
Type A<Type>::Get()
{ return CDataMember; }
template <class Type>
void A<Type>::Set(Type x)
{ result = x; }
//-----

template <class Type>
Type Max(Type x, Type y)
{ return ((x>y)?x:y); }
int main(void) {
A<int>    i = 10, i1= 2, i2 = 0;
A<char>   c = 'A';
A<char*>  s1 = "qqqq";
A<char*>  s2 = "aaaa";
A<char*>  s = "";
    //cout << Max(i1.Get(), s1.Get()) << endl;
    cout << Max(s1.Get(), s2.Get()) << endl;
    cout << Max(i1.Get(), i.Get()) << endl;
    i2.Set(Max(i1.Get(), i2.Get()));
    cout << i2.result << endl;
    return 0;
}
```

Question 11: Operator Overloading and friends [10 marks]

Complex numbers are objects with two parts known as the "real" and "imaginary" parts both floating point numbers. For example, (2.0, 3.0). These numbers add and subtract like two-vectors, e.g., (2., 3.) + (-1., -1.) = (1., 2.) Unlike vectors it is possible to multiply complex numbers: (a, b) * (c, d) = (ac-bd, ad+bc). The absolute value is defined as follows:

$$|(a, b)| = \text{square-root}(aa+bb)$$

A) Suppose we define a class of complex numbers. Does it need a d-tor? Explain [1 mark]

B) Define class Complex with the methods mentioned above. Include an << method but no subtraction method. What would be the output of this C++ Program [7 marks]

C) Write a short main program to illustrate the use of the methods above. [2 marks]

Question 12: Logical Relation and Rules [10 marks]

A) Give Prolog rules for checking the length of a list of items. (There is no restriction on what a list item may be.) For example, if presented with the goal

?- length([a, b, c], 3). Prolog would respond with: yes

whereas

?- length([a, b], 3). would result in: no

You may assume that the first argument of "length" in the original goal does not involve partially unspecified lists like "[a, b|Rest]", and that the second argument is a non-negative integer.

B) Suppose that we represent sets as lists in Prolog. So the set {a,b,c} would be represented as [a, b, c]. Now assume we have a list in which there may be repeats, [a,d,a,c,g,c] e.g. Therefore, this list cannot be a set.

Compose Prolog rules to convert such a list to one which correctly represents the set of the list elements, {a,d,c,g}. By submitting the goal "setify([a,d,a,c,g,c], S)." we would expect Prolog to say yes with binding S = [a,d,c,g].

Question 13: [10 marks]

A) Consider the following Prolog program. Give one or more Prolog rules for solving goals like "male(X)." when added to the previous 13 rules. Obviously, one or more similar rules for goals like "female(X)." should be easy to compose and are not required. **You may NOT assume that member rule is available to you. [5 marks]**

```
parent(john, sally).
parent(jim, mike).
parent(carol, john).
parent(carol, sue).
parent(sally, jim).
parent(jim, bob).
sibling(X, Y) :- parent(Z, Y), parent(Z, X), X \== Y.
malelist([john, jim, mike, bob]).
femalelist([sally, carol, sue]).
mother(U, V) :- parent(U, V), female(U).
father(U, V) :- parent(U, V), male(U).
brother(U, V) :- sibling(U, V), male(U).
sister(U, V) :- sibling(U, V), female(u).
```

PICK ONE OF THE TWO FUNCTORS BELOW (SO EITHER B OR C) PLEASE CIRCLE THE ONE YOU CHOSE.

B) Write a rule that removes repeated elements from a list. [5 marks]

C) Describe what list operation "foo" performs. [5 marks]

```
append([], Ys, Ys).
append([X|Xs], Ys, [X|Zs]) :- append(Xs, Ys, Zs).
foo([], []).
foo([H|T], X) :- foo(T, R), append(R, [H], X).
```
