

Final exam CMPS 258  
Spring 2005  
Duration: 90 minutes



The following questions are regarding the C and C++ languages unless specified otherwise:

01. Which of the function declarations below cannot appear in the same source file along with the following declaration? `int foo(int x);` (1pt)

- (a) `int foo(int x, int y);`
- (b) `double foo(int y);`
- (c) `double foo(double z);`
- (d) all of the above
- (e) none of the above

02. In what way C++ differs from Java when it comes to binding of messages to methods? (2pts)

03. What makes an operator binary? (1 pt)

- (a) it produces a result that can be stored in binary
- (b) it requires two operands
- (c) it is composed of two characters
- (d) it has two different meanings depending on where it is used
- (e) it requires a boolean operand

04. Which of the following the keyword `const` cannot be used to indicate? (1 pt)

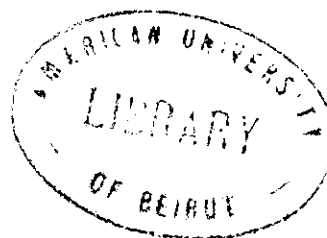
- (a) that a member function cannot change any of the member variables
- (b) that a variable cannot be changed
- (c) that a formal parameter cannot be changed
- (d) none of the above

05. In C++, given `y = 3`, the expression `(4 > y > 1)` (1 pt)

- (a) evaluates to `true`
- (b) evaluates to `false`
- (c) will not compile
- (d) throws an exception

06. Which of the function declarations below would not compile? (2 pts)

- (a) `int foo1(int x = 'A', int y = 2);`
- (b) `int foo2(int x, int y = 2);`
- (c) `int foo3(int x = 'A', int y = 'B');`
- (d) `int foo4(int x = 1, int y);`
- (e) none of the above



07. What is the output displayed by this code: (2 pts)

```
bool b = false;
if (b == true);
{
    cout << " Hello ";
}
if (b == false);
{
    cout << " Goodbye ";
}
```

- (a) Goodbye
- (b) Hello
- (c) Hello Goodbye
- (d) nothing is displayed

08. What is the output of the following C++ code? (1 pt)

```
int x = 1;
{
    int x = 2;
    {
        int x = 3;
        {
            int x = 4 ;
            cout << x << endl;
        }
    }
}
```

- (a) 1
- (b) 2
- (c) 3
- (d) 4
- (e) compiler error

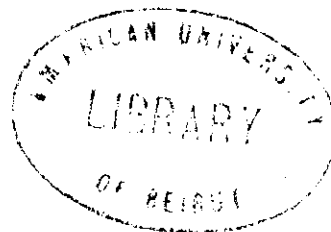
09. What is the output of the following Java code? (1pt)

```
int x = 1;
{
    int x = 2;
    {
        int x = 3;
        {
            int x = 4 ;
            System.out.println("x") ;
        }
    }
}
```

- (a) 1
- (b) 2
- (c) 3
- (d) 4
- (e) compiler error

10. What is the difference, if any, between a subclass and a subtype:

- a) In Java? (2 pts)
- b) In C++? (2 pts)





11. If you need to compare the type names of 2 objects in Java you would call `getClass().getName()` on the objects. How would you do the same in C++? (1pt)

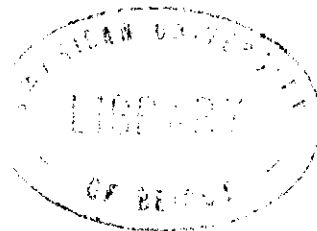
12. Provide the missing code in `MyClass` so that the following code compiles and prints out the word "Hello". Explain this behavior. (4pts)

```
class MyClass
{
public:
    char *data;
```

```
};
```

```
void foo(MyClass obj)
{
    printf("%s", obj.data);
}
```

```
void main()
{
    char str[] = "Hello";
    foo(str);
}
```



13. The function foo does not compile. Explain why, clearly. (3pts)

```
class MyClass
{
public:
    int x;
    const int foo() const {
        x++;
        return x;
    }
};
```

14. Given the following code: (3pts)

```
class MyClass
{
public:
    MyClass() { x = 1; }
    int x;
    int & bar()
    {
        return x;
    }
};

void main()
{
    MyClass obj;
    obj.bar() = 2;
}
```



- a) At the end of main, the value of x is 1
- b) At the end of main, the value of x is 2
- c) At the end of main, the value of x is 'junk'
- d) The code does not compile

15. What is the output of this code? (6 pts)

```
class B1
{
public:
    B1(){ data = 1; }
    B1(int data){ this->data = data; }
    int data;
};
class B2
{
public:
    B2(){ data = 2; }
    B2(B2 &b2) { }
    int data;
};
class A
{
public:
    A(int data1, int data2) {
        b1.data = data1;
        b2.data = data2;
    }
    B1 b1;
    B2 b2;
};
void main()
{
    A a1(100, 200);
    A a2(a1);
    A a3 = a1;
    cout << a2.b1.data << " " << a2.b2.data
         << " " << a3.b1.data << " " << a3.b2.data << endl;
}
```

- a) 100 200 100 200
- b) 100 200 1 2
- c) 1 2 100 200
- d) 100 junk 100 junk
- e) compiler error

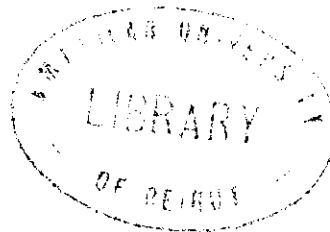
16. The following code does not compile. Describe and explain any error reported by the compiler.(2 pts)

```
class Vehicle {};
class Plane: public Vehicle {};

void pilot(Vehicle &v) {}
void fly(Plane &p) {}

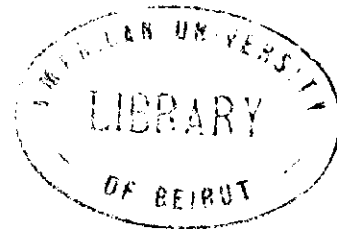
void main() {
    Vehicle v;
    Plane p;

    pilot(v); // line 1
    pilot(p); // line 2
    fly(v);   // line 3
    fly(p);   // line 4
}
```

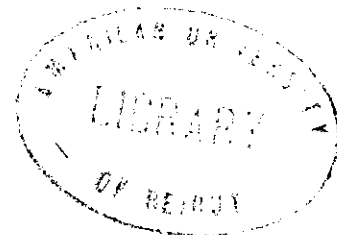


17. The following code does not compile. Describe and explain any error reported by the compiler. (2 pts)

```
class Vehicle {};  
class Plane: private Vehicle {};  
  
void pilot(Vehicle &v) {}  
void fly(Plane &p) {}  
  
void main() {  
    Vehicle v;  
    Plane p;  
  
    pilot(v); // line 1  
    pilot(p); // line 2  
    fly(v);   // line 3  
    fly(p);   // line 4  
}
```



18. When an object gets deleted in C++, its destructor gets called. Is there an equivalent concept in Java? Expand. (2 pts)



19. Do you see any potential problem in this code? If yes, provide a fix. (4 pts)

```
class Array {
public:
    Array(int *arr, int len) { // assume len > 0 and arr points to
        data = new int[len]; // allocated memory >= len*sizeof(int)
        for (int i = 0; i < len; i++)
            data[i] = arr[i];
    }
    Array(int i) {
        data = new int;
        *data = i;
    }

    ~Array() { delete [ ] data; }
private:
    int *data;
};
```



20. Given the C++ code (5pts)

```
class A {
public:
    A() {
        a = new int[3];
        for (int i = 0; i < 3; i++) a[i] = i;
    }
    ~A() { delete[] a; }

private
    int *a;
};

void init(A &x)
{
    A y;
    x = y;
}

int main()
{
    A p;
    ...
    init(p);
    ...
}
```



you might notice that as the program runs following the call `init(p)`, the values stored in `p.a` start changing unpredictably.

- a) What might be happening?
- b) Explain how to fix this problem and provide a modified code.

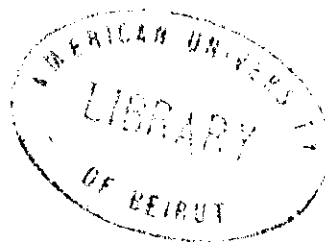


21. Complete this C++ program so that it will produce the output shown below. (15 pts)

Output:

Eagle: can fly, cannot swim  
Robin: can fly, cannot swim  
Owl: can fly, cannot swim  
Crow: can fly, cannot swim  
Woodpecker: can fly, cannot swim  
Ostrich: cannot fly, cannot swim  
Kiwi: cannot fly, cannot swim  
Swan: can fly, can swim  
Duck: can fly, can swim  
Penguin: cannot fly, can swim

```
class Bird {  
    string name;  
public:  
    Bird() { }
```





```

void printInfo( ) {
    cout << name << ": ";
    if (canFly( ))
        cout << "can fly, ";
    else
        cout << "cannot fly, ";
    if (canSwim( ))
        cout << "can swim";
    else
        cout << "cannot swim";

    cout << endl;
}
};
class Flightless_Bird:
public:
    Flightless_Bird( ) { }

```



```

};
class Swimming_Bird:
public:
    Swimming_Bird( ) { }

```

```
};
```

```

class Flightless_Swimming_Bird:
public:

```

```
};
```



```

int main( ) {
    Bird *bird[10] = {
        new Bird("Eagle"), new Bird("Robin"),
        new Bird("Owl"), new Bird("Crow"), new Bird("Woodpecker"),
        new Flightless_Bird("Ostrich"), new Flightless_Bird("Kiwi"),
        new Swimming_Bird("Swan"), new Swimming_Bird("Duck"),
        new Flightless_Swimming_Bird("Penguin")
    };
    for (int k=0; k<10; k++) bird[k]->printInfo( );
}

```

22. Consider the following C++ program:

```

class Node {
public:
    virtual int trace( ) { return 0; }
};
class Leaf: public Node {
    int value;
public:
    Leaf(int i): value(i) { }
    virtual int trace( ) {
        cout << "Leaf " << value << endl;
        return value;
    }
};
class NonLeaf: public Node {
    Node *left, *right;
public:
    NonLeaf(Node *a, Node *b): left(a), right(b) { }
    virtual int trace( ) {
        int i=left->trace( ); int j=right->trace( );
        int k=fun(i,j); cout << k << endl; return k;
    }
    virtual int fun(int, int) { return 0; }
};
class Add: public NonLeaf {
public:
    Add(Node *a, Node *b): NonLeaf(a,b) { }
    virtual int fun(int a, int b) {
        cout << "Add "; return a+b;
    }
};
class Subtract: public NonLeaf {
public:
    Subtract(Node *a, Node *b): NonLeaf(a,b) { }
    virtual int fun(int a, int b) {
        cout << "Subtract "; return a-b;
    }
};
class Multiply: public NonLeaf {
public:
    Multiply(Node *a, Node *b): NonLeaf(a,b) { }
    virtual int fun(int a, int b) {
        cout << "Multiply "; return a*b;
    }
};

```



```
class Divide: public NonLeaf {
public:
    Divide(Node *a, Node *b): NonLeaf(a,b) { }
    virtual int fun(int a, int b) {
        cout << "Divide "; return a/b;
    }
};
```

a) Draw the associated class inheritance hierarchy. (3 pts)



b) Draw the virtual table for each class. (6 pts)



(continued) Given the following client code:

```
int main( ) {
    Node *t=new Divide(new Leaf(14), new Leaf(7));
    Node *u=new Multiply(new Leaf(9), new Leaf(10));
    Node *v=new Subtract(new Leaf(13), new Leaf(11));
    Node *w=new Add(new Leaf(8), new Leaf(12));
    Node *x=new Subtract(u,t);
    Node *y=new Multiply(w,v);
    Node *z=new Add(y,x);
    z->trace( );
}
```

c) Draw the corresponding binary tree then show the output of the code. (10 pts)



23. Few minutes after this program starts running, the system slows down dramatically. Explain what might be happening and provide a fix. (6 pts)

```
class A
{
private:
    char *dataA;
public:
    A(){ dataA = new char[1000]; }
    ~A(){ delete [] dataA; }
};

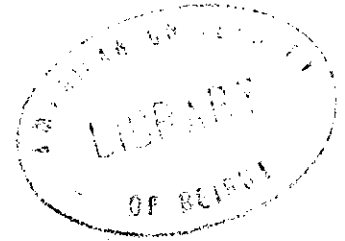
class B: public A
{
private:
    char *dataB;
public:
    B(){ dataB = new char[1000]; }
    ~B(){ delete [] dataB; }
};

void main()
{
    int count = 10000000;
    while (count > 0)
    {
        A *arr[] = {new A(), new B()};
        delete arr[0];
        delete arr[1];
        count--;
    }
}
```



24. What's wrong with this implementation? Give an example of when things will go wrong. (4 pts)

```
class String {
private:
    char *data;
public:
    String(const char *value) { ... } // not relevant
    ~String() { ... } // not relevant
    String& operator=(const String& rhs) {
        delete [] data;
        data = new char[strlen(rhs.data)+1];
        strcpy(data, rhs.data);
        return *this;
    }
};
```



25 Given the following class definition:

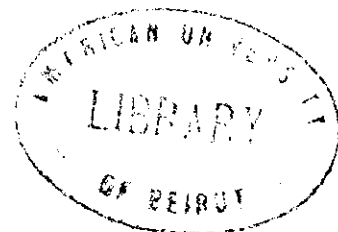
```
class Rational {
public:
    Rational(int numerator = 0, int denominator = 1);
    ~Rational();
private:
    int n, d; // numerator and denominator

    friend const Rational& operator*(
        const Rational& lhs, const Rational& rhs) {
        Rational result(lhs.n * rhs.n, lhs.d * rhs.d);
        return result;
    }
};

void main()
{
    Rational two = 2; // line 1
    Rational four = two * two; // line 2
}
```



- a) line 1 does not cause any compiler error. Why? (2 pts)
- b) Things can go wrong at line 2. Explain. (2pts)



26. If you are overloading the assignment operator in a class, what should you return? Explain. (2pts)

