

**Instructions:**

- Print your name in the space provided below.
- Answer each question in the space provided.
- If you want partial credit, justify your answers briefly and concisely, even when justification is not explicitly required.
- There are 15 questions, priced as marked. The maximum score is 100.
- When you have completed the test, sign the pledge at the bottom of this page and turn in the test.
- This is a closed-book, closed-notes examination. No calculators or other electronic devices may be used during this examination.
- You may not discuss (in any form: written, verbal or electronic) the content of this examination with any student who has not taken it. You must return this test form when you complete the examination. Failure to adhere to any of these restrictions is an Honor Code violation.

Do not start the test until instructed to do so!

Name _____
printed

Pledge: On my honor, I have neither given nor received unauthorized aid on this examination.

signed

Consider the following class declaration, (*slightly modified from the spring 2001 first test*):

```
enum Color {RED, GREEN, BLUE, YELLOW};
enum Direction {UP, DOWN, LEFT, RIGHT};

class CPacMonster {
private:
    Clocation mLoc;
    Color      mColor;
    bool      mActive;
public:
    CPacMonster( ); // Line 1
    CPacMonster(Clocation L, Color C); // Line 2
    CPacMonster& setColor(Color C = RED); // Line 3
    Color getColor() const; // Line 4
    CPacMonster Move(Direction Dir, int Distance = 1); // Line 5
    Clocation getLocation() const; // Line 6
    ~CPacMonster(); // Line 7
};
```

Assume the following object declarations are in scope:

```
CPacMonster Doggett, SmokingMan;
```

1. [7 points] For the given statement below very briefly explain the effect of the execution of the statement upon the data members of the object Doggett.

```
Doggett.Move(LEFT).setColor(YELLOW);
```

2. [7 points] For the given statement below very briefly explain the effect of the execution of the statement upon the data members of the object SmokingMan.

```
SmokingMan.setColor(RED).Move(LEFT, 1);
```

3. [4 points] If the Clocation object in the CPacMonster class was declared:

```
private:
    Clocation* mLoc;
```

Assuming that no other changes are made to the class interface, would the composition be static or dynamic?

For the next 4 questions, consider the following classes:

```

class BoxCar {
private:
    string Label;
    Crate** Cargo; //ptr to array of ptrs
    int Size;
    int numCars;
public:
    BoxCar(string L = "None",
            int Sz = 10);
    BoxCar(const BoxCar& RHS);
    bool addCrate(Crate*& newCrate);
    BoxCar& operator=(
        const BoxCar& RHS);
};

BoxCar::BoxCar(string L, int Sz) {
    Label = L;
    numCars = 0;
    if (Sz <= 0) {
        Size = 0;
        Cargo = NULL;
    }
    else {
        Size = Sz;
        Cargo = new Crate*[Size];
    }
}

BoxCar::BoxCar(const BoxCar& RHS) {
    // implementation not shown
}

bool BoxCar::addCrate(Crate*& newCrate) {
    if (numCars == Size)
        return false;
    Cargo[numCars] = newCrate;
    numCars++;
    newCrate = NULL;
    return true;
}

BoxCar& BoxCar::operator=(const BoxCar& RHS) {
    // implementation not shown
}

class Crate {
private:
    string Label;
public:
    Crate(string L = "None");
    string getLabel() const;
};

Crate::Crate(string L) {
    Label = L;
}

string Crate::getLabel() const {
    return Label;
}

```

4. [7 points] Consider execution of the following code fragment:

```

for (int I = 0; I < 10; I++) {
    BoxCar* B = new BoxCar(100);
    delete B;
}

```

Determine whether this code causes a memory leak. If yes, explain clearly how the leak occurs. If no, explain clearly what prevents a leak from occurring.

5. [7 points] Does the class `BoxCar` need a destructor? If not, explain why not. If yes, write an implementation of the destructor.

6. [7 points] List all class member functions that are invoked in executing the following code:

```
BoxCar B1("Fred", 10), B2;  
B2 = B1;
```

7. [7 points] List all class member functions that are invoked in executing the following code:

```
BoxCar B1("Fred", 10);  
BoxCar B2 = B1;
```

Consider the following classes:

```
class Operand {  
private:  
    int Op;  
public:  
    Operand(int V = 5) {Op = V;}  
    int getOp() const {return Op;}  
};
```

```
class Sum {  
private:  
    Operand RHS, LHS;  
    int S;  
public:  
    Sum();  
    // irrelevant fns not shown  
};  
  
Sum::Sum() {  
    S = RHS.getOp() + LHS.getOp();  
};
```

8. [7 points] Is the relationship between `Sum` and `Operand` an association or something else? Justify your answer.

Consider the following class (which uses the `BoxCar` and `Crate` classes declared earlier):

```
class Train {
private:
    int numCars;
    BoxCar* Cars[100];
public:
    Train();
    bool addCar(BoxCar* B);
    BoxCar* removeCar();
};

Train::Train() {
    numCars = 0;
    for (int I = 0; I < 100; I++)
        Cars[I] = NULL;
}

bool Train::addCar(BoxCar* B) {
    if (numCars == 100)
        return false;
    Cars[numCars] = B;
    numCars++;
    return true;
}

BoxCar* Train::removeCar() {
    if (numCars == 0)
        return NULL;
    BoxCar* T = Cars[numCars];
    Cars[numCars] = NULL;
    numCars--;
    return T;
}
```

9. [7 points] Is the relationship between `Train` and `BoxCar` an association or something else? Justify your answer.

Consider the following class:

```
class Sentence {
private:
    string* Words;
public:
    Sentence(ifstream& In, int N);
    ~Sentence();
};

Sentence::Sentence(ifstream& In, int N) {
    if ( N > 0 ) {
        Words = new string[N];
        // input code omitted
    }
    else Words = NULL;
}

Sentence::~Sentence() {
    delete [] Words;
}
```

10. [7 points] Is the relationship between `Sentence` and `string` an association or something else? Justify your answer.

11. [5 points] In C++, when an object is used as an actual parameter and passed to a function by reference, the formal parameter is:

- 1) a copy of the actual parameter, made by the assignment operator.
 - 2) a copy of the actual parameter, made by the copy constructor.
 - 3) logically the same object as the actual parameter.
 - 4) This is not allowed.
 - 5) None of these
-

Consider the description below of an alarm clock:

The Big Ben company came up with a clock that design experts still call one of the best ever, the Moon Beam. The Moon Beam clock flashes a gentle blinking alarm light for four minutes before the chime alarm sounds. Features a streamlined moonbeam-yellow case and genuine glass face with illuminated dial, snooze function, easy-to-read numerals, replaceable 25-watt bulb and a built in battery backup for power outages. 5"H, 6½"W, 2"D.

[7 points each] Choosing from the following answers,

object class attribute behavior none

in terms of designing an object-oriented model of the Moon Beam alarm clock determine whether each of the entities listed below is best characterized as a(n) _____ in the system, or if it is none.

12. alarm

13. the built in battery

14. blinking

15. 25-watt