CSI 32 SPRING 2024 REVIEW PROBLEMS (Ojakian)

NOTE: A bunch of example problems follow in no particular order. These practice problems (some easier, some hard, some not explained so clearly!). The final exam will be strongly based on this! It is essential tha you study and work through this. This is NOT for a grade; just practice

1. Write an expression for each part (that is one line of code which evaluates to the desired value).
   (a) Suppose x has been declared to be an int. Write an expression that evaluates to 5 times the value of x; the value of x should not change.
   (b) Suppose x has been declared to be an int. Write a line of code which updates the value of x to be 5 times its current value.
   (c) Write an expression that evaluates to true exactly when the value of the integer variable y is NOT negative.
   (d) Suppose y has been declared to be an int. Write a line of code which updates the value of y to be 0.

2. (a) Write a line which declares variable X to be a vector of double.
   (b) Write a line which declares variable A to be an array of 50 integers.
   (c) How many bytes of memory does the array A occupy?

3. (a) Write a line of code that declares p to be a pointer to int.
   (b) Write a line of code that makes p point to the integer x.
   (c) Suppose `GOOD` is a class. Declare a variable named `X` as a pointer to `GOOD`.

4. Write a C++ function which takes no inputs and returns nothing. It just **prints** "Hello".

5. Write a C++ function which takes no inputs and just **returns** the integer 42.

6. What is the value of y after running the following code?

```
int y = -2;
int x = y;
y = y + 5;
y++;
x = x + 20;
```

7. Write a C++ program that gives the same print out as the following Python program (do it using a for-loop and then a while loop).

```
A = 1
for i in range(100):
    print(A)
    A = A + 5
```

8. Indicate exactly what the following program prints.

```
int main() {
    int i;
    vector<int> A = {2, 16, 0, 2};

    for (i = 3; i >= 0; i=i-1) {
        cout << "First: " <<  i * 2 << endl;
        cout << "Second: " << A[i] - 2 << endl;
    }

}
```

9. Indicate exactly what the following program prints.

```
int main() {
    int k = 8;
    int n = 14;
    while (k < n) {
        cout << k << endl;
        k = 2 + k;
     }

     cout << "Final k: " << k << endl;
     cout << "Final n: " << n << endl;
}
```

10. Write a C++ program that reads in an integer input from the user (using "cin") and then prints out "NEG" if the integer is negative, prints "EVEN" if the integer is positive and even, prints "ODD" if the integers is positive and odd, and prints "ZERO" if the integer is 0.

11. Use a for-loop to write a C++ program that prints the multiples of 5 from 100 through 1000 (including both 100 and 1000).

12. Use a while loop to write a C++ program that asks the user for integers until they enter a negative integer. Then print out the product of all the non-negative integers given by the user.

    For example: If the user inputs 4, 2, 4, 1, -7, then your program should print out 32 (i.e. $4*2*4*1$).

13. Enhance the last program. Have it do the product, but also print out the string of products.

    For example: If the user inputs 4, 2, 4, 1, -7, then your program should print out the STRING:

    $$4*2*4*1 = 32$$

14. An integer is prime if it is larger than 1 and the only positive integers dividing into it evenly are 1 and itself (for example: 7 is prime, but 9 is not prime). Write a C++ program that gets an integer from the user. Then it prints out "PRIME" if the integer is prime, and "NOT PRIME" if the integer is not prime.

15. Write a function called `factor` which takes a single positive integer as input and returns any factorization of it - choose some reprentation, for example, say as a vector of length 2. If it is prime, just return back the given number.

16. (HARD?) Write a function called `totalFactor` which takes a single positive integer as input and returns its complete factorization as a string.

    Example: `totalFactor`(20) returns "2*2*5"

17. (a) Write a statement that declares a prototype for a function `CUNY`, which has two parameters. The first is a reference to int and the second is a pointer to double. The function returns an int.

    (b) Suppose `BAD` is a class. Declare a variable named `B` as a an object of type `BAD` and `P` as a pointer to `BAD`.

18. Write a C++ function `increasing` that takes one array (of ints), and an integer n as input and returns a bool: true if within the first n entries of the array, the integers are strictly increasing; returns false otherwise.

    Now modify the program to replace "stricly increasing" by "non-decreasing" (i.e. increases, but being the same is ok).

19. Write a C++ function `isReverse` that takes two arrays (of ints), and an integer n as input and returns a bool: true if one is the reverse of the other, false otherwise.

    Assuming you have the `isReverse` function, write a simple function (without loops) that returns a bool for whether or not the array is a palandrome.

20. Write a C++ function `reverse` that takes one array (of ints), and an integer n as input and returns the array in reverse.

21. Write a C++ function `prime` that takes an int input and returns a bool: true if the number is prime and false otherwise.

    Then use the `prime` function to write a program that sums all the primes up to and including some integer n.

22. (a) Write the definition of a function which takes no inputs and returns no outputs, and just prints 3 X's.

(b) Write the definition of a function which takes no inputs and returns no outputs, and just prints 100 X's. Do this 1) with a for loop, 2) with a while loop.

(c) Do the last problem, but now the function takes an int argument and prints that many X's.

23. (a) Write a line which declares variable *V* to be a unordered map from int to strings.

(b) Declare a variable *x* to be an integer type, and initialize it to -1.

(c) Write a statement that declares a prototype for a function `CSI`, which has two input parameters. The first input is an int and the second is a pointer to string. The function returns nothing.

24. What does the following code print?

```
for (int y = 0; y < 4; y++) {
    cout << (y * 2) << " ";
    cout << (y + 2) << endl;
}
```

25. Write the interface in order to declare a class `Clock` containing:

- A data member `time` of type int.
- A constructor that accepts no arguments.
- A method called `increment` that accepts one int parameter and returns no value.
- A method called `getValue` that accepts no parameters and returns an int.

26. Write appropriate methods for the last `Clock` class, so that it runs on 24 hours. That is increment increases time by 1, but instead of going to 24 it cycles back to 0.

27. If the C++ function `FF` is defined as below, what are the values of the following:

(a) `FF(2,-3)`

(b) `FF(467, -1) + 9`

(c) `FF( FF(1,2) , 0 )`

```
int FF(int x, int y) {
    x++;
    y++;
    return x*y;
}
```

28. Write a function which takes a vector of integers as input and returns the product of the numbers in the vector. Do it again, but make one that ignores zeros in the vector.

29. Write the definition of the class `Square` that has:

- One data member `length`, of type double.
- The constructor, that takes one double as input (its length).
- Function member `setLength` that takes one input of type double, and sets the length of the square to this value.
- Function member `perimeter` that returns the perimeter of the square.

30. Modify the last `Square` class to make it appropriate for a Rectangle.

31. Suppose we have defined the following class `Test2`:

```
class Test2 {
public:
    Test2(int y) {
        x = y;
    }

    int get() {
        return x;
    }

    void mod() {
        x = x - 1;
    }
    // declaring internal data
    int x;
};
```

Suppose we have also defined the following function `GG`:

```
int GG(Test2 T) {
    T.mod();
    T.mod();
    return T.get();
}
```

What does the following program print?

```
int main() {
    Test2 X(10);
    Test2 Y(0);
    cout << "Before: " << X.get()<< ", " << Y.get() << endl;
    a = GG(X);
    b = GG(Y);
    cout << "After: " << X.get()<< ", " << Y.get() << endl;
    cout << "a: " << a << ", b: " << b << endl;
}
```

32. The following uses the class `Test2` from the last problem. After running the following piece of code, that is the value of `kk`? And what is the value of the internal data `x` in `AA` and in `BB`?

```
Test2 AA(100);
Test2 BB(400);
int kk = 0;
while (AA.get() < BB.get()) {
    BB.mod();
    kk++;
}
```

33. What does the following C++ program print?

```
int Fun(int x, vector<int> V) {
     x = x + 5;
     int y = V[2];
     V[2] = 42;
     return x + y;
}

int main() {
     int x = 10;
     vector<int> V = {3, 4, 7, 8};
     int y;

     y = Fun(x, V);

     cout << "x: " << x << endl;
     cout << "y: " << y << endl;
     cout << "V[2]: " << V[2] << endl;
}
```

34. Write a C++ function named `OPP` which takes a string as input. If the input string is "+" then the function returns "-". If the input string is "-" the function returns "+". If the input string is anything else, then the string "0" is returned. If your code is correct, it should print as indicated in the comments below:

```
cout << OPP("+"); // Should print: -
cout << OPP("-");  // Should print: +
cout << OPP("ANYTHING Else");  // Should print: 0
```

35. Use the last `OPP` function to write a function `negate` which takes a string as input where the string is an integer written as a string, with the first symbol being a "+" or "-" unless it is "0". For example, some inputs are: "+48" or "-7" or "0". The function should return a string which is the negated number; for example on the previous inputs, it would return: "-48" or "+7" or "0".

36. Suppose n has been declared to be an int, and n is initialized to some positive value. Write a C++ program which finds the sum of the even integers between 1 and n, including n. Then it prints out the sum. For example: If $n = 8$, then your program should print out 20, since 2 + 4 + 6 + 8 = 20.

37. Consider the Python function below named `pyth`. You are going to write two C++ functions that produce the same output as `pyth`, for all integer inputs. You are asked to do this in two ways:

    (a) Write a C++ program (call it `cppA`) that produces the same output as `pyth`. It **must** use loops, and may **not** use any conditional statements.

    (b) Write a C++ program (call it `cppB`) that produces the same output as `pyth`. It may **not** use loops, but may use conditional statements.

    For example, `pyth`(5) returns 10, so both of your C++ functions should return 10 on input 5. As another example (don't forget the negative integers!), `pyth`(-5) returns 0, so both of your C++ functions should return 0 on input -5.

```
def pyth(x):
    A = 0
    for i in range(x):
        A = A + 2

    return A
```

38. Write a C++ function `swap` that takes two references to ints as input, and swaps the values. Repeat this exercises, but now the function takes two pointers to double, and swaps the contents (but does not change where the pointers point - i.e. does not change the address of the pointers

39. For this problem we do addresses in decimal. Suppose p is a pointer to int and the current address stored at p is 1500 (remember pointers store an address). Suppose the data at address 1500 is 45. Suppose q is a pointer to int and the current address at q is 2000. Suppose the data at address 2000 is 99. What is the result of each of the following (for each assume the starting point just stated), i.e. what are the contents of p and q, and what address is stored at p and q after code execution? In C++ how would you get the actual address?

    (a)
```
int x;
int y;
int *p = &x;
int *q = &y;
*p = *q;
```

    (b)
```
int x;
int y;
int *p = &x;
int *q = &y;
p = q;
```

40. Write a C++ function named `FACTOR` which takes one int variable as pass-by-reference. After `FACTOR` is called and its execution is complete, the variable passed in should be the result of dividing it by its largest factor less than it.

For example, if written correclty, your code should print out as indicated in the comments below.

```
int x = 20;
FACTOR(x);
cout << x; // should print 2 since the largest factor of 20 (less than 20) is 10, and 20/10 = 2.
```

Question: Describe the integers for which FACTOR will not change its value?

41. Indicate exactly what the following program prints out, justifyng your answer.

```
int main() {
    int A = {2, 4, 8, 16};
    int *p;
    p = A;
    cout << "First: " << *p << endl;

    p = p + 2;
    p++;
    cout << "Second: " << *p << endl;

    p = A;
    cout << "Third: " << *p << endl;

    for (int i = 0; i <= 2; i++) {
        A[i] = A[ (i+1) ];
    }

    p--;
    p = p + 3;
    cout << "Fourth: " << *p << endl;
}
```

42. Write a function `crazy` which takes two reference arguments to int, where the second is a const reference. If the second argument is odd, then double the first argument, and otherwise subtract 1 from the first argument.

Suppose the value of *x* is 5, and the value of *y* is 6. How many times should `crazy`(x, y) be called before x is zero?

43. What does the following C++ code print. You must justify your answer fully and clearly to get any credit! At the end of the program there is a comment requesting you to add lines of code; add the appropriate lines of code next to that comment.

```
int main() {
    int a = 2;
    int b = 5;
    int *X;
    int *Y;
    X = &a;
    Y = &b;

    *Y = *X;
    *X = *X + 2;
    cout << "First X: " << *X << endl;
    cout << "First Y: " << *Y << endl;
    cout << "First a: " << a << endl;
    cout << "First b: " << b << endl;

    Y = X;
    *X = *X + 2;
    cout << "Second X: " << *X << endl;
    cout << "Second Y: " << *Y << endl;
    cout << "Second a: " << a << endl;
    cout << "Second b: " << b << endl;

    Y = new int;
    *Y = 0;
    *X = *X + 2;
    cout << "Third X: " << *X << endl;
    cout << "Third Y: " << *Y << endl;
    cout << "Third a: " << a << endl;
    cout << "Third b: " << b << endl;

    // ADD HERE: ANY NEEDED LINES OF CODE TO AVOID A MEMORY LEAK
}
```

44. Suppose we have defined a C++ function as follows:

```
int Fun(int a, int &b, int *c) {
     a = b;
     b = *c;
     *c = b;
     return a;
}
```

Then indicate exactly what the following program prints out, justifying your answers.

```
int main() {
    int aOut = 10;
    int bOut = 20;
    int x = 30;
    int *cOut = &x;

    int y;
    y = Fun(aOut, bOut, cOut);

    cout << aOut << endl;
    cout << bOut << endl;
    cout << *cOut << endl;
    cout << x << endl;
    cout << y << endl;
}
```

45. Consider a function with the following prottype:

```
int GG(const vector<int> &V1; vector<int> V2, vector<int> &V3);
```

Describe the differences in the three ways a vector of ints is passed to the function, giving an example function to illustrate these differences.

46. In the following C++ program, note the comment in the third line. Supposing that line really does print out as that line indicates, describe what the rest of the program prints. Then answer the questions below the program. You must justify your answers!

```
int main() {
    double A[4] = {2, 9, 0, 2};
    cout << &A[0] << endl;  // Suppose it prints (when converted to decimal): 5000

    for (int i = 1; i < 4; i++) {
        cout << "next: " << &A[i] << endl;
    }
}
```

Put what program prints:

Suppose the array of integers were extended, but still, the address of $A[0]$ is 5000 (in decimal). Answer the following:

(a) What is the address of A[100]?

(b) Which bytes does the data at A[100] occupy?

(c) If the array were an array of ints, instead of doubles, and $A[0]$ is at address 5000, then what is the address of A[100]?

47. Write a C++ function `opposite` that takes two arrays (of ints), and an integer n as input and returns a bool: true if among the first n entries, if one array has an even integer at an entry, then the other has an odd entry; and returns false otherwise. You may assume that both arrays are of length at least n.

For example, `dom`(A, B, 3) should return true if A is the array $\{3, 4, 3, 9\}$ and B is the array $\{4, 7, 6, 21\}$, since among the first three entries, if one array has an even integer, the other has an odd.

48. Write a C++ function named `ZERO` which takes one int variable as a reference variable. After `ZERO` is called and its execution is complete, the variable passed in should be 0. **For example**, if written correclty, your code should print out as indicated in the comments below.

```
int x = 20;
ZERO(x);
cout << x; // should print 0.
```

49. What does the following code print?

    Also, at the end of the program there is a comment requesting you to add lines of code; add the appropriate lines of code next to that comment.

```
int main() {
    int a = 2;
    int b = 5;
    int *X;
    int *Y;
    X = &a;
    Y = &b;

    *Y = *X;
    *X = *X + 2;
    cout << "First X: " << *X << endl;
    cout << "First Y: " << *Y << endl;

    Y = X;
    *X = *X + 2;
    cout << "Second X: " << *X << endl;
    cout << "Second Y: " << *Y << endl;

    Y = new int;
    *Y = 0;
    *X = *X + 2;
    cout << "Third X: " << *X << endl;
    cout << "Third Y: " << *Y << endl;

    // ADD HERE: ANY NEEDED LINES OF CODE TO AVOID A MEMORY LEAK
}
```

50. Indicate exactly what the following program prints out.

```
int main() {
    int A[4] = {9, 7, 6, 2};
    int *p;
    p = A;
    cout << "First: " << *p << endl;
    p = p + 2;
    cout << "Second: " << *p << endl;

    p = A;
    p++;
    cout << "Third: " << *p << endl;
}
```

51. Suppose we have defined a function as follows:

```
int Fun(int a, int &b, int *c) {
     a = a + 2;
     b = b + 2;
     *c = *c + 2;
     return a;
}
```

Then indicate exactly what the following program prints out.

```
int main() {
    int aOut = 10;
    int bOut = 20;
    int x = 30;
    int *cOut = &x;

    int y;
    y = Fun(aOut, bOut, cOut);

    cout << aOut << endl;
    cout << bOut << endl;
    cout << *cOut << endl;
    cout << x << endl;
    cout << y << endl;
}
```

52. In the following program, note the comment in the third line. Supposing that line really does print out as that line indicates, describe what the rest of the program prints. Then answer the questions below the program.

```
int main() {
    double A[4] = {2, 9, 0, 2};
    cout << &A[0] << endl;  // Suppose it prints the Hex address: 0x61fdd0

    for (int i = 1; i < 4; i++) {
        cout << "next: " << &A[i] << endl;
    }
}
```

Put what program prints:

Answer the following:

(a) How many total bytes does the array occupy?

(b) What is the value stored at A[3]?

(c) Which bytes does the data at A[3] occupy?

53. Write the interface of a class `Recipe` meant to simulate a cooking recipe, containing:

- Two data members: One is a vector of strings and the other is a vector of doubles.
- A constructor that accepts no arguments.
- A method called `addIngredient` that accepts one string parameter, one double parameter and returns no value.
- A method called `getQuantity` that accepts one string parameter and returns a double.
- A method called `numIngredients` that accepts one string parameter and returns an int.

54. Write the implementation of the `Recipe` class from the previous question. The class contains:

- The data members mentioned above
- A constructor that accepts no arguments and initializes the vectors to be empty.
- A method called `addIngredient` that accepts one string parameter, one double parameter and returns no value; it should pair up the given string with the given amount, so that if `getQuantity` is called with that string, it returns that amount.
- A method called `numIngredients` that accepts no parameters. It returns the number of ingredients

55. Write a template function that takes an input and prints it out 5 times.

56. Write a template function that takes any input as its first argument and an int as its second argument (say n), and prints it out the first argument n times.

57. Suppose an array A has been declared and n is its length. Write some code to shift all the values one place to the right, and make it so the last element of the array becomes the first element of the array.

    If this operation is done twice, how many more times should it be done to get back to the original array?

    Give an example of an array of length 8 for which one of these shifts produces a different array, but two of these shifts produces the same array you started with.

58. Write a template class for an Ordered Set of any kind of item: By this we mean that you can push and pop items as usual, but push will only do something it that item is not already in the list. It should be template, meaning that the item could be int, string, double, etc. Suggestion: Use vector as your internal data for this template class.

59. Write a function that takes a positive integer n as input and returns a length n vector of ints so that the entries are increasing powers of 2, i.e 2, 4, 8, 16, ... , and so on.

60. Write a function, which when called does this: Repeatedly ask the user for a number (using cin) till 0 is entered, then return the sum of the numbers up to that point.

61. Suppose p1, p2, and p3 are all pointers to int.

    (a) Write some code that leaves the pointers pointing to the same places, but updates the contents so that p1 contains the contents of p3, p2 that of p1, and p3 that of p2.

    (b) Do the last problem, but now change it so that you change where each points, for example, p1 points to p3, etc.

    (c) Print out the addresses in both cases to verify that you have followed the instructions.

62. Write a class Person and a class Student which is a derived class of Person. Fill in some reasonable methods ....

63. Suppose A is an array of short ints at address $6fe$ (in hex) and is of length 5. What is the address of $A[4]$? What are the addresses of the bytes it occupies?

    Suppose p is a pointer to short int and we set $p = A$, and then execute: $p = p + 3$; then what are the following: $*p$, p, sizeof(*p), sizeof(A).

64. State what is wrong with the following code:

```
const double pi = 3.14;
pi++;
```

65. Write a *template function* called `temp` which takes two inputs of the same type and returns the larger of the two inputs.

    Example: `temp`(3, 9) returns 9, while `temp`(4.5, 1.25) returns 4.5 (i.e. the one template function can handle different types of inputs - ints, doubles, etc).

66. Write the implementation of a class called `Num` that meets the following specifications:

    (a) It stores one `double` *D* as private data.

    (b) It has a constructor which takes one `double` argument as input, and initializes *D* to this value.

    (c) It has a method `inc` which takes one double as input and adds this input to the current value of *D*; this method returns nothing.

    (d) It has a method `half` which takes no inputs and returns nothing; it makes *D* to be half its value.

    (e) It has a method `display` which takes no inputs and returns nothing; it just prints out the current value of *D*.

    (f) It has a method `sign` which takes no inputs and returns a `bool`: it returns `true` if *D* is positive (i.e. bigger than zero), and `false` otherwise.

67. Consider the following two classes

```
class Parent {
public:
    Parent(int x) { data = x; }
    int get() { return data; }
    void mod(int x) { data = x; }
    void up(int x) {data = data + x; }
    // Add any new methods to Parent here
int data;
};

class Child : public Parent {
public:
    Child(int x) : Parent(x) { }
    void mod(int x) { data = 2*x; }
    // Add any new methods to Child here
};
```

What does the following code print?

```
Parent P(100);
Child C(5);
cout << P.get() << endl;
cout << C.get() << endl;
P.mod(50);
C.mod(10);
cout << P.get() << endl;
cout << C.get() << endl;
```

Add a method `up` to the class Child (write in the above space in the Child class) and add a method `down` to just one of the classes (*either to the Child class or the Parent class; not both! Write in the appropriate space above next to the class Child or the class Parent*), so that the following lines print out as indicated in the comments.

```
Parent P2(1000);
Child C2(200);
P2.up(5);
C2.up(10);
cout << P2.get() << endl; // Should print: 1005
cout << C2.get() << endl; // Should print: 220
P2.up(10);
C2.up(20);
cout << P2.get() << endl; // Should print: 1015
cout << C2.get() << endl; // Should print: 260
P2.down(15);
C2.down(100);
cout << P2.get() << endl; // Should print: 1000
cout << C2.get() << endl; // Should print: 160
```

68. Consider the following complete program, which consists of the class `Dynamic`, then the function `Dfunct`, then a `main()`.

    (a) Indicate exactly what the program prints, in exactly the order it prints it (*the order of the prints is important*).

    (b) Fill in the one missing line of code which should be in the destructor.

```
class Dynamic {
public:
    Dynamic(int x) {
        cout << "In Constructor" << endl;
        p = new int;
        *p = x;
    }

    ~Dynamic() {
        cout << "In Destructor" << endl;
        // FILL IN APPROPRIATE LINE HERE
    }

    Dynamic(const Dynamic &source) {
        cout << "In Copy Constructor" << endl;
        p = new int;
        *p = *(source.p);
    }

    void operator=(const Dynamic &source) {
        cout << "In Assignment Operator" << endl;
        *p = *(source.p);
    }

    void display() { cout << "Value: " << *p << endl; }

private:
    int *p;
};

void Dfunct(Dynamic D) {
    D.display();
}

int main() {
    Dynamic D1(11);
    Dfunct(D1);

    Dynamic D2(22);
    D2 = D1;
    D2.display();
}
```

69. Suppose we have the following classes (questions on next page!)

```
class RPGchar {
    private:
        int hp;
        string al;

    public:
    RPGchar(int hitpoints, string alignment = "good") {
        hp = hitpoints;
        al = alignment;
    }
    int getHP() { return hp; }
    bool isAlive() { return (hp > 0); }
    void improve(int x) { hp += x; }
    void damage() {
        hp -= 5;
        if (hp < 0) {
            hp = 0;
        }
    }
};

class Wizard : public RPGchar {
    public:
    Wizard(int hp, string al = "evil") : RPGchar(hp, al) {
    damage();
    }
};

class Fighter : public RPGchar {
    public:
    Fighter(int hp, string al = "evil") : RPGchar(hp, al) {
    improve(10);
    }
};
```

70. Refer to classes on last page. What does the following print.

```
int main() {
    RPGchar R(4, "evil");
    cout << "R: " << R.getHP() << endl;
    Wizard W(8);
    cout << "W: " << W.getHP() << endl;
    Fighter F(15);
    cout << "F: " << F.getHP() << endl;

    for (int i = 0; i < 6; i++) {
        W.improve(3);
        F.damage();
    }

    cout << "W: " << W.getHP() << endl;
    cout << "W state: " << W.isAlive() << endl;
    cout << "F: " << F.getHP() << endl;
    cout << "F state: " << F.isAlive() << endl;
}
```

71. Write a method override for damage in the Wizard class so that if she hits 0, he bounces back to having one hit point.

72. Override improve for Figher, so that on input x he goes up twice x.

73. Modify the classes so that there is a maximum number of hitpoints which you cannot go beyond. The max for the parent class is 100, for the Wizard is 80, and for the Figher is 120. Modify the methods appropriately.

74. Consider following class and function (questions on next page).

```
class Simple {
public:
    Simple(int v = 0) {
        cout << "In Constructor with x = " << v << endl;
        x = v;
        p = new int;
        *p = x+1;
    }

    ~Simple() {
        cout << "In Destructor; x = " << x << endl;
        delete p;
    }

    Simple(const Simple &source) {
        cout << "In Copy Constructor with source.x = " << source.x << endl;
        x = source.x;
        p = new int;
        *p = *(source.p);
    }

    void operator=(const Simple &source) {
        cout << "In Assignment Operator with source.x = " << source.x << endl;
        x = source.x;
        *p = *(source.p);
    }

    bool operator<(const Simple &right) { return (x < right.x); }
    int getX() { return x;}
    int getP() { return *p; }
    void setX(int v) { x = v; }
    void setP(int v) { *p = v; }

private:
    int x;
    int *p;
};

void F(Simple U) {
    cout << "x in F: " << U.getX() << endl;
    U.setX(99);
    U.setP(2021);
}
```

75. What does the following print? (use class on last page)

```
int main() {

    Simple S(5);
    cout << "x: " << S.getX() << endl;
    cout << "p: " << S.getP() << endl;

    Simple Sbig(8);
    Sbig.setP(2);
    cout << "Bool: " << (S < Sbig) << endl;

    S.setX(9);
    S.setP(20);
    cout << "x: " << S.getX() << endl;
    cout << "p: " << S.getP() << endl;

    cout << endl;
    F(S);
    cout << "x: " << S.getX() << endl;
    cout << "p: " << S.getP() << endl;

    cout << endl;
    cout << "Assignments" << endl;
    Simple S2;
    S2 = S;

    cout << endl;
    S = S;

    cout << "S2 x: " << S2.getX() << endl;
    cout << "S2 p: " << S2.getP() << endl;

    cout << endl;
    cout << "Pointers to object" << endl;
    Simple *Sptr;
    Sptr = &S;
    cout << "Sptr x: " << Sptr->getX() << endl; // abbreviate (*Sptr).getX()
    cout << "Sptr p: " << Sptr->getP() << endl;
    Sptr = new Simple(7);
    Sptr->setP(77);
    cout << "Sptr x: " << Sptr->getX() << endl;
    cout << "Sptr p: " << Sptr->getP() << endl;
    delete Sptr;
    cout << "\nAT END" << endl;
}
```