

CS 115 Final Review Quiz

Rules

- You must briefly explain your answers to receive partial credit.
- When a snippet of code is given to you, you can assume that the code is enclosed within some function, even if no function definition is shown. You can also assume that the `main` function is properly defined and that the `iostream`, `fstream`, `iomanip`, `string`, `cstring`, and `cmath` libraries have been included at the beginning of the program.
- When you are asked to write *a snippet* of code, you may also assume that it is enclosed within some function that any necessary libraries have been included.
- When you are asked to write *a complete program*, you must write the `#include` statements, the `int main()`, etc. in your solution for full credit.
- A line consisting solely of “...” represents one or more unspecified C++ statements, some of which may change the values of program variables.

Problem 1: 15 points.

Match the following descriptions with the term they describe by writing that term in the space provided. The choices of terms are:

- this
- constructor
- destructor
- class
- object
- pointer
- overloading
- operator

Not all terms will be used.

(a) When two functions have the same name but different input parameters

(b) A function that is automatically called when an object is created

(c) A variable whose data type is a class

(d) A variable containing the address of another variable

(e) A pointer to the object whose member function is being called

Problem 2: 15 points.

Using the selection sort code on the next page, which works identically to the code from your labs, show the contents of the following 5-element array after each iteration of SelectionSort's loop in the labeled boxes. *Leave the boxes blank if the loop does not complete a given iteration.*

INITIAL VALUE

8	6	1	2	3
---	---	---	---	---

AFTER i=0

--	--	--	--	--

AFTER i=1

--	--	--	--	--

AFTER i=2

--	--	--	--	--

AFTER i=3

--	--	--	--	--

AFTER i=4

--	--	--	--	--

AFTER i=5

--	--	--	--	--

AFTER i=6

--	--	--	--	--

Code for Problem 2.

You may tear this page out of your exam.

```
void SelectionSort(int arr[], int size) {
    int min_pos = 0;

    // This is the loop in question
    for (int i = 0; i < size-1; i++) {
        min_pos = FindMinPos(arr, i, size);
        if (min_pos != i) {
            swap ( arr[i], arr[min_pos] );
        }
    }
}

int FindMinPos(int arr[], int start, int size) {

    int minVal = arr[start];
    int minPos = start;

    for (int i = start+1; i < size; i++) {
        if (arr[i] < minVal) {
            minVal = arr[i];
            minPos = i;
        }
    }
    return minPos;
}

void Swap (int& a, int& b) {
    int temp = a;
    a = b;
    b = temp;
}
```

Problem 3: 15 points.

Complete the definitions of the following functions as indicated by their comments. You may assume that any necessary libraries have been included.

Hint: Each function can be written in two lines of code or fewer, none of which are `cin` or `cout` statements.

(A)

```
/* Return the sum of the lengths of the two input
   C-strings. */
int Funct2 (char* a, char* b) {
```

```
}
```

(B) For this part, assume that the class `Rectangle` has private fields

```
int length;
int width;
```

You will be writing the `>` operator.

For your reference, the copy constructor for the class is given below:

```
/* Copy constructor for Rectangles. */
Rectangle::Rectangle (const Rectangle& other) {
    this->length = other.length;
    this->width = other.width;
}

/* Overloaded ">" operator for the Rectangle class.
   Returns TRUE if the area of this is greater than the
   area of other and FALSE otherwise.
   You fill this in. */
bool Rectangle::operator > (const Rectangle& other) const{
```

```
}
```

Problem 4: 15 points.

For this problem, you must write a **class definition** for a class named **Grades** that contains the following:

Note that data members should be private and member functions should be public.

For now, you are only writing prototypes for the member functions. You will define the functions in the next problem.

- A student's name (as a `string`)
- A student's test average and homework average (as two variables of type `double`)
- Prototype for a default constructor
- Prototype for a function called `SetGrades`. This function will take two variables of type `double` as inputs and will return a `bool`.
- Prototype for a function called `SetName`. This function will take a `string` as input and will not return anything.
- Prototype for a function called `GetAvg`. This function will return the student's overall average based on his/her test average and homework average.
- Prototype for a function called `IsPassing`. This function will return `true` if the student's overall average is passing and `false` otherwise.

Problem 5: 20 points.

In this problem, you will write definitions for the functions in the class `Grades`. Here is a little bit more information about the functions. You may assume that `<string>` is included. *Note that none of your code for this problem should include `cin` or `cout` statements!*

- The default constructor will initialize the name to "" (the empty string), the test average to 0, and the homework average to 0.
- The `SetGrades` function will work as follows:
 - If one or both of the inputs is less than zero or greater than 100, it will return `false`.
 - Otherwise, it will set the test average equal to the first input and the homework average equal to the second input. Then it will return `true`.
- You do not have to write the `SetName` function, but you can assume in Problem 7 that someone else has provided it.
- The `GetAvg` function will return the student's overall average. Tests are 60% of the overall grade, and homework is 40% of the overall grade. In other words:
$$\text{Overall Grade} = 0.6 * (\text{test average}) + 0.4 * (\text{homework average})$$
- The `IsPassing` function will return `true` if the student's overall average is greater than or equal to 60, and `false` otherwise. This function should call `GetAvg` in order to compute the average, rather than duplicating code.

Problem 6: 20 points.

Assume that the class definitions you wrote in Problems 5 and 6 are located in a file called `grades.h` in the same directory as the program you're about to write.

Write a **complete program** that uses the `Grades` class from `grades.h` to do the following:

- Create one `Grades` object with the student name "Alice", a test average of 75, and a homework average of 100.
- Create another `Grades` object with the student name "Bob." Ask the user to supply Bob's test and homework averages and update the object with that information. You can assume that the user's input is valid.
- Using the member function `isPassing`, print out the number of students who are passing the class (0, 1, or 2).

REFERENCE

C-string functions:

<code>strlen</code>	Input is a C-string. Returns the length of the string (not including the null terminator). Usage example: <code>length = strlen(name);</code>
<code>strcat</code>	Input is two C-strings. Appends the second string to the end of the first string (the first string is changed, but the second is not). Usage example: <code>strcat(str1, str2);</code>
<code>strcpy</code>	Input is two C-strings. Copies the second string to the first string, overwriting the original contents. Usage example: <code>strcpy(str1, str2);</code>
<code>strcmp</code>	Input is two C-strings. Returns 0 if they are the same, a negative number if <code>str2</code> is alphabetically greater than <code>str1</code> , and a positive number if <code>str1</code> is greater than <code>str2</code> . Usage example: <code>if(strcmp(str1, str2) > 0)</code>