CSCI 3232 Systems Software Assignment 4

Upload all your files to the associated dropbox in Folio before the deadline --- **11:30PM Feb 19, Tuesday, 2019.**

**Note:** Make all your codes compilable and runnable under Ubuntu. Do not put your codes in Word or PDF documents. Make them separate source files (.h, .c, .cpp etc) as you would compile them. **Include a Makefile to compile all your programs. Make sure you have tested that your makefile works.**

1. (20 points) What is the output of the following C++ program? You should be able to tell the output **without** running the program.

#include <iostream>

using namespace std;

class programming {

protected: int variable;

public:

programming() {

cout << "In constructor\n";

input\_value(100);

}

programming(int a) {

cout << "In constructor\n";

input\_value(a);

}

~programming() {

cout << "In destructor\n";

output\_value();

}

void input\_value(int b) {

cout << "In function input\_value\n";

variable = b;

}

void output\_value() {

cout << "Variable is "<< variable << "\n";

}

void onemorefunction() {

if(variable%2) cout<< "Variable is odd\n";

else cout<< "Variable is even\n";

}

};

programming ob;

int main(int argc, char \*argv[])

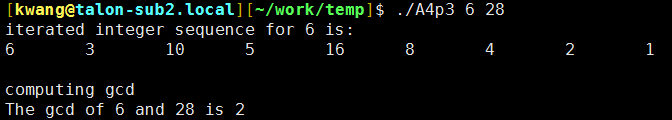
{

programming object(3);

object.onemorefunction();

return 0;

}

1. (35 points) Write a C++ program A4p2.cpp with a class of your own design. The class should contain a protected **int** member variable *var*, which is initialized with an integer value between 1 and 50 in a constructor that takes an integer parameter. The class should contain a public member function called *play* that should print out a sequence of integers as a result of iteratively applying a math function *f* to the member variable *var*. The function *f* is defined as f(x)=3x+1 if x is odd and f(x)=x/2 if x is even. Stop the iteration when the value 1 is reached. (Example: When *var* is 6, the *play* function’s output sequence should be 6,3,10,5,16,8,4,2,1.) In your main function create an object of this class whose member *var* should be initialized with the first command line argument to your program (that is, argv[1]) and then call the *play* member function to output the sequence of desired integers. You can check whether the supplied first command line argument is an integer between 1 and 50 in your main function. A sample run can look like the following picture. Submit source code A4p2.cpp. No screen shot is needed.
2. (35 points) Write a C++ program A4p3.cpp with a class that is a derived class of your class from problem 2. Add a private **int** member variable *var2* to this class. Initialize the member variables *var* and *var2* with values between 1 and 50 using a constructor that takes two integer parameters. Add a public member function called *getgcd* that should print out the greatest common divisor of *var* and *var2*. (You can refer to gcd.c in Folio’s sample codes directory for how to compute the greatest common divisor of two integers.) In your main function, create an object of this class, initialize its members *var* and *var2* with the first two command line arguments to your program (that is, argv[1] and argv[2]) and then call the *play* and *getgcd* functions. A sample run can look like the following picture. Submit source code A4p3.cpp. No screen shot is needed. Please do **NOT** just submit A4p3.cpp because it includes all functionality of A4p2.cpp. You need to submit two separate program source files.
3. (10 points) What is the output of the following C++ program? You should be able to tell the output **without** running the program.

#include <iostream>

using namespace std;

int main(int argc, char \*argv[])

{

int arr[2][3]={{25,16,7},{58,49,30}};

int \*b[2]={arr[0],arr[1]};

cout<<arr[0][1]-\*(\*(arr+1)+2)<<endl;

cout<<b[0][1]-\*(\*(b+1)+2)<<endl;

return 0;

}

1. Note: You need to supply a single **makefile** to compile your programs in both problems 2 and 3. Without a correct makefile, up to 5 points will be deducted. Do as requested: do **not** combine programs 2 & 3 into one program just because they share some parts. As stated in the last slide of 3\_Pointers\_Functions.pptx, it is OK to rename your makefile as makefile.txt before uploading it to Folio.

|  |  |
| --- | --- |
| 1 | In constructor  In function input\_value  In constructor  In function input\_value  Variable is odd  In destructor  Variable is 3  In destructor  Variable is 100 |
| 2 | See code |
| 3 | See code |
| 4 | -14  -14 |