McMaster University
MECHTRON 2MD3: Data Structures and Algorithms for Mechatronics

Winter 2023
Assignment 1
Due: Jan. 22, 2023 at 23:59

Instructions

All questions annotated "short answer" should be typed and submitted in a single pdf document with the file name "1234567-asg1.pdf" where 1234567 is your student ID. Each question annotated "programming" should be submitted as a single C++ source file named "1234567-asg1-x.cpp" where 1234567 is your student ID and x is the question number. All code in programs and short answers should be formatted and commented. Please ensure your source files compile and run properly before submitting.

Questions

1. [short answer, 3 marks]: Declare A to be a pointer to integer and assign a value of 21 to its referent. How would you write an expression whose value is twice the value of A's referent?

2. [short answer, 2 marks]: Consider the following attempt to allocate a 10-element array of pointers to doubles and initialize the associated double values to 0.0. Rewrite the following (incorrect) code to do this correctly. (Hint: Storage for the doubles needs to be allocated.)

```
double* dp[10];
for (int i = 0; i < 10; i++) dp[i] = 0.0;
```

3. [short answer, 1 mark]: What (if anything) is different about the behaviour of the following two functions f and g that increment a variable and print its value?

```
void f(int x)
{ std::cout << ++x; }

void g(int& x)
{ std::cout << ++x; }
```

4. [short answer, 4 marks]: Write a short C++ function that takes a positive double value *x* and returns the number of times we can divide *x* by 2 before we get a number less than 2.

5. [short answer, 4 marks]: The greatest common divisor, or GCD, of two positive integers *n* and *m* is the largest number *j*, such that *n* and *m* are both multiples of *j*. Euclid proposed a simple algorithm for computing GCD(*n, m*), where $n > m$, which is based on a concept known as the Chinese Remainder Theorem. The main idea of the

algorithm is to repeatedly perform modulo computations of consecutive pairs of the sequence that starts (*n, m*,...), until reaching zero. The last nonzero number in this sequence is the GCD of *n* and *m*. For example, for *n* = 80,844 and *m* = 25,320, the sequence is as follows:

```
80,844 mod 25,320   =   4,884
25,320 mod 4,884    =   900
4,884 mod 900       =   384
900 mod 384         =   132
384 mod 132         =   120
132 mod 120         =   12
120 mod 12          =   0
```

So, GCD of 80,844 and 25,320 is 12. Write a short C++ function to compute GCD(*n, m*) for two integers *n* and *m*.

6. [programming, 10 marks]: The birthday paradox says that the probability that two people in a room will have the same birthday is more than half as long as the number of people in the room (*n*), is greater than 23. This property is not really a paradox, but many people find it surprising. Design a C++ program that can test this paradox by a series of experiments on randomly generated birthdays, which test this paradox for *n* = 5,10,15,20,...,100. You should run at least 100 experiments for each value of *n. Your program should* output a single comma-separated line for each *n showing: 1) the value of n;* 2) the number of experiments that returned two people in that test having the same birthday; 3) the *measured* probability of 2 people in the group having the same birthday. To calculate "measured probability" for each *n*: let *c* be the number of experiments in which at least 2 people had the same birthday and let *e* be the number of experiments. We define the "measured probability" as c/e. Example output is as follows:

```
5,4,0.04
10,18,0.18
15,33,0.33
// multiple lines omitted...
90,100,1
95,100,1
100,100,1
```

7. [short answer, 2 marks]: Suppose we have a variable *p* that is declared to be a pointer to an object of type Progression using the classes of Section 2.2.3. Suppose further that *p* actually points to an instance of the class GeomProgression that was created with the default constructor. If we cast *p* to a pointer of type Progression and call *p*->nextValue(), what will be returned? Why? Please assume that the nextValue() function is public.

8. [programming, 8 marks]: Write a short C++ program that creates a Pair class that can store two objects declared as generic types. Demonstrate this program by creating and printing Pair objects that contain five different kinds of pairs, such as <int,string> and <float,long>. Your class should include a print function to display pairs in the format "<value1, value2>". An example main method testing 3 pairs and its associated output are below:

   Main method:

```
int main() {
  Pair<int, double> p1(1, 7.3);
  p1.print();

  Pair<std::string, double> p2("hello", 7.7);
  p2.print();

  Pair<float, long> p3(1.2, 777777773);
  p3.print();
}
```

   Output:

```
<1, 7.3>
<hello, 7.7>
<1.2, 777777773>
```

9. [programming, 5 marks]: Write a C++ class that is derived from the Progression class to produce a progression where each value is the absolute value of the difference between the previous two values. You can use this example from the 2md3_2023 git repo as a starting point: 2md3_2023/lecture_demos_ch02/polymorphism_demo.cpp You should include a default constructor that starts with 2 and 200 as the first two values and a parametric constructor that starts with a specified pair of numbers as the first two values. Include a main method that tests your class using both constructors and generating a progression of 10 values for each. Your program should output 4 lines like the following:

```
Absolute progression with default constructor:
200 198 2 196 194 2 192 190 2 188
Absolute progression with custom constructor:
300 297 3 294 291 3 288 285 3 282
```

The End.