

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

Assignment 1

Instructions

Each question annotated “programming” should be submitted as a single C++ source file named “1234-asg1-x.cpp” where 1234 is your student ID and x is the question number. Your solutions should be as implemented as efficiently as possible and with good C++ coding practices discussed in class. All code should be formatted and commented. Please ensure your source files compile and run properly before submitting.

Question 1

[programming, 5 marks]: It’s 2050, and the price of a single potato is now \$30. A store initially has 20 potatoes and tax is still 13%.

Part A) Write a function “BuyOnePotato” that decrements the number of available potatoes by 1, where:

Arguments: potato price, and number of currently available potatoes

Returns: money that should be paid including tax.

Part B) Write a function “BuyNumPotatoes” that also decrements the number of available potatoes by N, where N is the number of potatoes the customer wants to purchase, where:

Arguments: potato price, number of currently available potatoes, and the number of potatoes to be purchased.

Returns: money that should be paid including tax.

Both parts should be done without the use of global variables. Your functions will be tested with a main method like this, which you should include in you source file submission:

```
int main() {
    int potatoes_available = 20;
    float potato_price = 30.0;

    std::cout << "$" << BuyOnePotato(potato_price, potatoes_available)
               << std::endl;
    std::cout << potatoes_available << std::endl << std::endl;
    std::cout << "$" << BuyNumPotatoes(potato_price, potatoes_available, 3)
               << std::endl;
    std::cout << potatoes_available << std::endl;
    return EXIT_SUCCESS;
}
```

Question 2

[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 $n = 5, 10, 15, 20, \dots, 100$. 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
```

Your implementation must:

- Include a Birthday class which represents valid birthdays with month and day, and implements a method to randomize the date.
- Overload the “==” operator to test if two birthday objects are equal.
- Use the following interface to implement a Birthday paradox checker:

```
class BirthdayParadox {
public:
    Birthday *birthdays;    // Array for holding birthdays
    const int num_people;    // Number of birthdays to generate
    BirthdayParadox(int n);  // Constructor with n random birthdays
    ~BirthdayParadox();      // Destructor
};
```

- Finally, your BirthdayParadox class must implement a method named “CheckDuplicates” which returns 1 if a duplicate birthday is found in a group of birthdays and otherwise returns 0;

Question 3

[programming, 5 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:

```
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>
```

Question 4

[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_2025 git repo as a starting point: 2md3_2025/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.