

This test has 2 pages. Please write your name, the course number and “Test 1” on the exam booklet.

Books, notes, calculators, computers and phones are *not* permitted. Work on the questions in the order that you want, but try to write your answers in order in the booklet.

If a question asks for C++ code, don’t worry about remembering every little detail of C++ syntax. Minor details will not affect your grade.

You can keep your copy of the exam. Or leave it in the booklet and you’ll get it back.

1. (20%) Answer each of the following questions briefly but precisely.

- (a) What is a test driver?
- (b) What is abstraction?
- (c) What is a constructor?
- (d) What are the two main advantages of C++ strings over C strings?

2. (64%) Create a class `Product`. Each object in this class represents a product. Each product has a number, a description and a price. The number is a positive integer, the description is a string and the price is a real number. Include the following operations. Whenever appropriate, declare methods to be inline. (Don’t forget to implement the methods.)

- (a) A default constructor that initializes the product to have number 0, description *none* and price -1 .
- (b) A constructor `Product(number, description, price)` that initializes the product to the given values. The arguments are of type **int**, **string** and **double**, respectively.

- (c) A method `price()` that returns the price of the product.
- (d) A method `set_description(new_description)` that sets the description of the product to the given value. The argument is of type `string`.
- (e) A method `print` that prints the product to standard output (`cout`) in the following format:

```
Number, Description: Price
```

The price should be preceded by a dollar sign (\$). Assume that all prices already have exactly two digits after the decimal point.

- (f) An input operator (`>>`) that reads a product from an input stream. Assume that the number, description and price of the product are given on separate lines, in that order:

```
Number
Description
Price
```

No error-checking is performed.

- (g) A less-than operator (`<`) that returns **true** if the number of the first product is less than the number of the second product.

3. (16%) Implement the C string function `strcat(dest, source)`. Recall that this function adds a copy of C string `source` to the end of C string `dest`. The function simply assumes that the array that holds `dest` is large enough.