CSc10300 - Assignment 4

Object and Classes/ Pointers/ Streams - Fall 2021

Motahare Mounesan

1

Create a class named Line:

(a) Define private data members p1 and p2 as pointer to Point objects (the one we had in lectures), slope and length as double variables.

(b) Define setter and getter functions.

(c) Define a null-constructor that initializes numeric variables with zero and allocate dynamic memory for points and initialize them to [0,0] as well.

(d) Overload a constructor that allocates memory for points, initilize them with given arguments, and calculate the slope and length.

(e) Implement destructor, copy constructor and copy assignment operator.

(f) Create a function called "parallel" that takes too Line objects, returns true when given lines are parallel and returns false otherwise.

(g) Overload the less than (<) and greater than (>) and equality (==) operators (compare the length).

(h) Write a functions that reads lines in the format provided in the lines.txt from the file (**without any change**) and stores them in a vector named Lines.

(i) Sort the objects of Lines vector in ascending order using the *sort* function from *algorithm* library.

(j) Extend the functionality of **cin** and **cout** for this class. For cin function you need to define a template like [(x1,y1,),(x2,y2)], and ask user to enter the entire input at once. Then you will parse the input, extract x1, y1, x2, and y2 and create an object of Line with them.

(k) Extensively test your code as you learned in unit testing lessens. Make sure you cover border cases.

You should be able to decide where (in which file) to define each of these functions based on what you have learned so far.

2

Assume the Product structure is declared as follows:

```
struct Product
{
    string description;  // Product description
    int partNum;         // Part number
    double cost;         // Product cost
};
```

(a) Add two constructors to the Product structure declaration. The first should be a default constructor that sets the description member to the null string and the partNum and cost members to zero. The second constructor should have three parameters: a string, an int, and a double. It should copy the values of the arguments into the description, partNum, and cost members.

(b) Define a **print** function as member of the struct that prints an object of this struct in the following format.
Description: Claw Hammer Part
Number: 547
Part Cost: $8.29

(c) Declare a dynamic array of Products of size 5 and name it "items". Then initilize it with user input values.

(d) Write a **print** function (not as a member of the struct) and pass a pointer to the pointer that points to the array (double pointer) and print all the items of the array in that function.

(e) Define a **max** function (not as a member of the struct) that gets an array of items as an input and returns a pointer to the max element of the array.

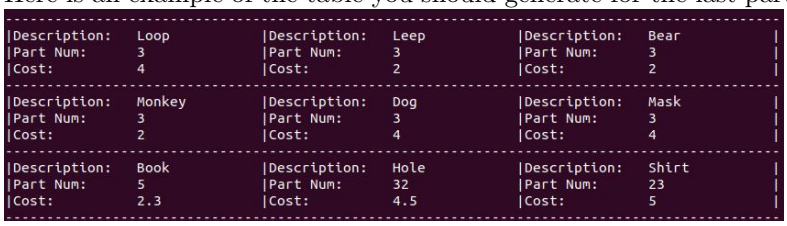
(f) Declare a 3 by 3 two dimensional **dynamic array** and populate it.

(g) Define an **output** function that takes a stream object and a pointer to a 2D array as arguments and outputs data members of objects in format of 3*3 table into the given stream. Test your function both with an output file stream and cout stream.

(h) Write a testbench to test your program properly.

Submission:

1. Here is an example of the table you should generate for the last part of h.



Description:	Loop	Description:	Leep	Description:	Bear
Part Num:	3	Part Num:	3	Part Num:	3
Cost:	4	Cost:	2	Cost:	2
Description:	Monkey	Description:	Dog	Description:	Mask
Part Num:	3	Part Num:	3	Part Num:	3
Cost:	2	Cost:	4	Cost:	4
Description:	Book	Description:	Hole	Description:	Shirt
Part Num:	5	Part Num:	32	Part Num:	23
Cost:	2.3	Cost:	4.5	Cost:	5

2. In your unit testing, test the steps of the problem in order.