

cis1.5-spring2007-sklar, lab III

instructions

- This assignment will be worth 9 points.
- **Both parts together are due on Monday March 26** and must be submitted by email (as below).
- **Follow these emailing instructions:**
 1. Create a mail message addressed to *sklar@sci.brooklyn.cuny.edu* with the subject line **cis1.5 hw3**.
 2. Attach **ONLY** the **.cpp** files for each part, as outlined below.
DO NOT ATTACH THE **.cbp** (CodeBlocks Project) files!
 3. Failure to follow these instructions will result in points being taken away from your grade. The number of points will be in proportion to the extent to which you did not follow instructions... (which can make it a lot harder for me to grade your work — grrrr!)

description.

For this assignment, you will write two programs that simulate the behavior of a manufacturing robot. Pretend that the robot works in a factory and knows how to make three types of parts: part A, part B and part C. A user tells the robot to make a part, and the robot randomly generates one of these three parts.

In the first program, your user will tell the robot how many parts to make altogether, and the robot will generate that number of total parts. Then the program will output to the screen a list of how many of each type of part the robot made. Here is an example run:

```
how many parts would you like the robot to generate? 34
generating parts: C A C C A B B A A B C B A C B B C A C A B B B C A B
B A B A B A A B
number of A parts: 12
number of B parts: 14
number of C parts: 8
```

In the second program, your user will tell the robot which part she wants and how many of that part, and the robot will keep generating parts until it has made enough of the one that the user requested. Then the program will output to the screen a list of how many of each type of part the robot made in total. Here is an example run:

```
which part would you like the robot to generate (A, B or C)? A
how many parts would you like the robot to generate? 5
generating parts: B B A A B B B C C C B B C C A B B C A C B A
number of A parts: 5
number of B parts: 10
number of C parts: 7
```

You will create the programs by writing a series of functions, described below, and then putting them all together with a `main()`.

part 1: generate total number of parts.

Create a new file called `parts1.cpp`.

- Write a function called `getPart()` that has no parameters and returns a `char`. The function should randomly choose a number (0,1,2) and then return a part (A, B, C) that corresponds to the randomly chosen number. For example, if the random number is 0, then the function should return A, etc. (1 point)
- Declare three global integer variables to keep count of how many of each type of part the robot generates (e.g., `int numPartA`). (1 point)
- Write a function called `displayParts()` that displays the value of each part count (i.e., the value of the three global variables declared above). (1 point)
- Write a function called `generateParts()` that takes one parameter: the total number of parts that you want the robot to generate (`int numParts`). This function should call `getPart()` to generate the total number of parts equal to its argument (`numParts`). As the robot generates parts, the function should count how many of each part it generates. (Hint: use a for loop.) (2 points)
- Write a `main()` function that does the following:
 1. ask the user how many parts she would like the robot to generate
 2. read the user's answer from the keyboard
 3. call the function `generateParts()` to generate the parts and keep a count of how many of each part is generated
 4. call the function `displayParts()` to display how many of each part was generated(1 point)

Compile, build and run your program to make sure it works as you expect it to.

part 2: generate a number of specific parts.

Create a new file called `parts2.cpp` and copy your `parts1.cpp` program into it.

- Modify the `main()` function so that it first asks the user which part she would like the robot to generate, and then asks the user how many of that part she would like. (1 point)
- Modify the function called `generateParts()` so that it takes two parameters: (1) a character variable containing the name of the part the user would like the robot to generate and (2) an integer variable containing the number of that part the user wants generated. (1 point)
- Further modify the function called `generateParts()` so that it generates parts *until* the total number of the requested part has been generated. (Hint: use a while loop.) (1 point)

Compile, build and run your program to make sure it works as you expect it to.