CS21A Introduction to Computing

Final Project Specifications

Sem 1 AY 2018-2019

Specifications

The final project builds off the midterm project with minor modifications. The Beneficiary class is no longer needed for this project.

A ReliefCenter receives donations of different relief goods. The ReliefCenter has a name and a collection of ReliefGood objects. Depending on the ReliefCenter's allocations, packs of ReliefGoods are created and released upon request. Each pack is composed of a predetermined number of units of each ReliefGood. The ReliefCenter keeps track of how many packs of ReliefGoods it has released in total.

ReliefCenter must be modified to **eliminate printing to or reading input from the console**. Methods now return strings which will be displayed on a Graphical User Interface (GUI) that you will create using Windows Forms.

Create a Windows Forms project that is associated with a single ReliefCenter object instance and facilitates transactions with it by calling methods whenever appropriate. For more details, refer to the documentation of ReliefCenter below.

An example of one possible GUI is shown below. **You have complete freedom in designing your GUI, as long as the functions required by this project are supported.** You may also include additional features which may net bonus points.

ReliefGood remains unchanged from the midterm project.

The maximum grade you can get by following the minimum requirements for this project is 86. To get a higher grade, implementation of **useful** additional features will be required (a list of which is also provided below).

Submission details

Include a **properly accomplished Individual Certificate of Authorship** in your project directory. Archive your **project directory** and name it **FinalProject-[Surname]-[IDnumber].zip** and submit it through Moodle by <deadline>.

This is an **individual project**. For any questions, contact your instructor.

Prepare for a final project defense during finals week.

class ReliefCenter

Constructor

ReliefCenter (string n)

- The constructor creates an instance of a ReliefCenter with a name. A new ReliefCenter also has a List<ReliefGood> for ReliefGoods. A new ReliefCenter begins with a total number of packs released of 0.
- Parameters
    - n – the name of the ReliefCenter

Methods

public string AddNewGood (string n, int r, string u)

- Enables the ReliefCenter to add new kinds of accepted ReliefGoods.
- Checks if the good is already in their list of accepted goods by calling the FindGood method.
- If the ReliefGood being added is already on the list, the ReliefGood is not added and a message saying the ReliefGood is already on the list of accepted ReliefGoods is returned.
    - Example: “Sorry, ReliefCenter already accepts rice.”
- if the ReliefGood being added is not on the list of accepted ReliefGoods, a new instance of Reliefgood is created and added to the list, and a message saying the Reliefenter now accepts the new ReliefGood is returned.
    - Example: “ReliefCenter now accepts rice.”
    - Example: “Added new good: rice.”
- Parameters:
    - n – name of the ReliefGood to be added
    - r – release rate
    - u – the unit of measurement
- Returns:
    - a string that reports the result of the operation

public ReliefGood FindGood (string g)

- Returns the ReliefGood object from the ReliefCenter’s list of accepted ReliefGoods with a name that matches the parameter g.
- Returns null if a match is not found.
- FindGood is **case-insensitive**.
- Parameters:
    - g – the name of the ReliefGood being searched
- Returns:
    - a ReliefGood object

public string GetName ( )

- GetName returns the name of the ReliefCenter

public int GetPackCount ( )

- Returns the total number of packs the ReliefCenter has released.

public string PrintInventory ( )

- Returns a string containing the status of all ReliefGoods in the ReliefCenter's inventory using the following format:

```
==== INVENTORY ====
Packs Released: 25
==================
rice: 100 kilos
water: 500 liters
sardines: 55 cans
==================
```

- The more ReliefGood objects are accepted, the more there is to report.

public string ReceiveGoods (string good, int num)

- This method enables the ReliefCenter to receive donations of a specific ReliefGood. It calls FindGood to check if the ReliefCenter accepts the ReliefGood being donated.
- If the ReliefGood is accepted, the number of units donated will be added to the supply of that ReliefGood, and a message is returned to acknowledge receiving the goods.
    - Example: "Received rice: 100 kilos"
    - Example: "ReliefCenter thanks you for your donation of 100 kilos of rice."
- If the ReliefGood is not accepted, the units will be rejected and a message is returned to inform the donor that the donation cannot be accepted.
    - Example: "Sorry, ReliefCenter does not accept diamonds."
    - Example: "ReliefCenter relief center does not accept Ma Ling at the moment."
- Parameters:
    - good – the name of the ReliefGood to be donated
    - num – the number of units to be donated
- Returns
    - a string that reports the result of the operation

public string ReleasePacks (int num)

- ReleasePacks releases one allocation of each ReliefGood that the ReliefCenter carries based on their respective release rates. Checks if all of the ReliefGoods in the list have enough units to supply the needed number of packs.
- If a lack of one ReliefGood is detected, the method stops checking the rest of the goods in the list. If there are not enough units of the ReliefGoods to supply the needed packs, no packs are released, and a message is returned to report the lack of supply.
    - Example: “Sorry, ReliefCenter does not have enough to supply 20 packs.”
- If there are enough units of all ReliefGoods for the needed number of packs, an allocation is released for each pack. An allocation refers to the corresponding release rate of each ReliefGood. A message is returned to report the successful release of all the requested packs.
    - Example: “ReliefCenter released 50 requested packs.”
- The method also updates how many packs of ReliefGoods the ReliefCenter has released in total.
- Parameters:
    - num – the number of packs needed
- Returns:
    - a string that reports the result of the operation

Notes and Tips:

- All input strings will be valid (not empty, alphanumeric, etc.)
- All input integers are positive
- All methods are case-insensitive
- You may use Windows Forms elements that were not discussed in class.
- Be careful when accessing variables from Forms elements. For example, label1.ToString() and label1.Text.ToString() return different values.
- Cite all your resources (URLs, books, dev blogs, etc.) in your Certificate of Authorship.
- Designing your Forms project from start to finish before putting in any code may help you visualize your project more clearly and make it less confusing.

Suggested extra features:

Here are a few suggested extra features.

You do not have to implement all of these to get a score of 100.

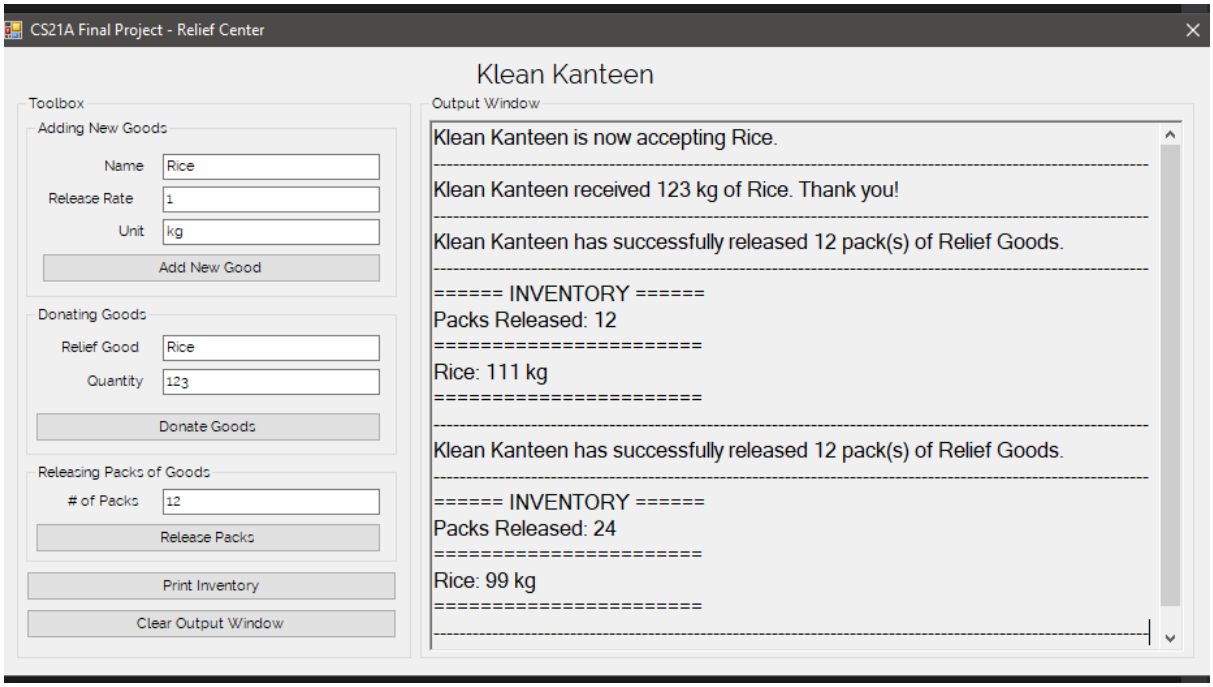
Extra features are not limited to this list.

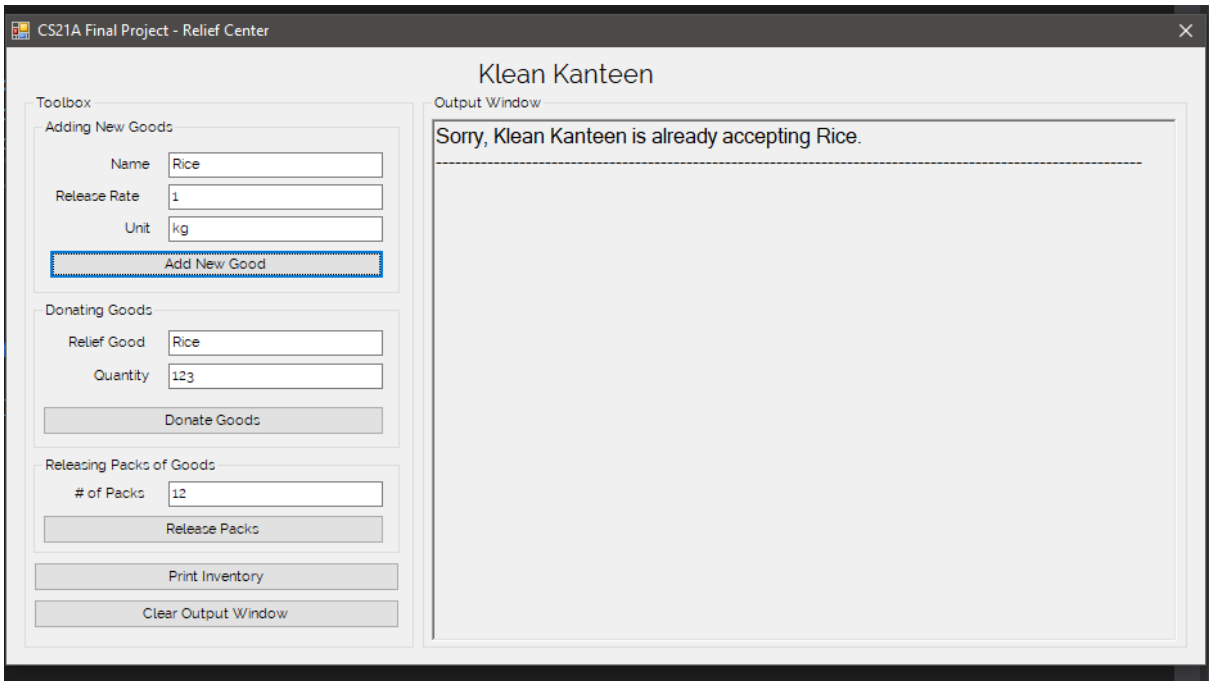
Bonus points will be awarded depending on the complexity of the implemented feature.

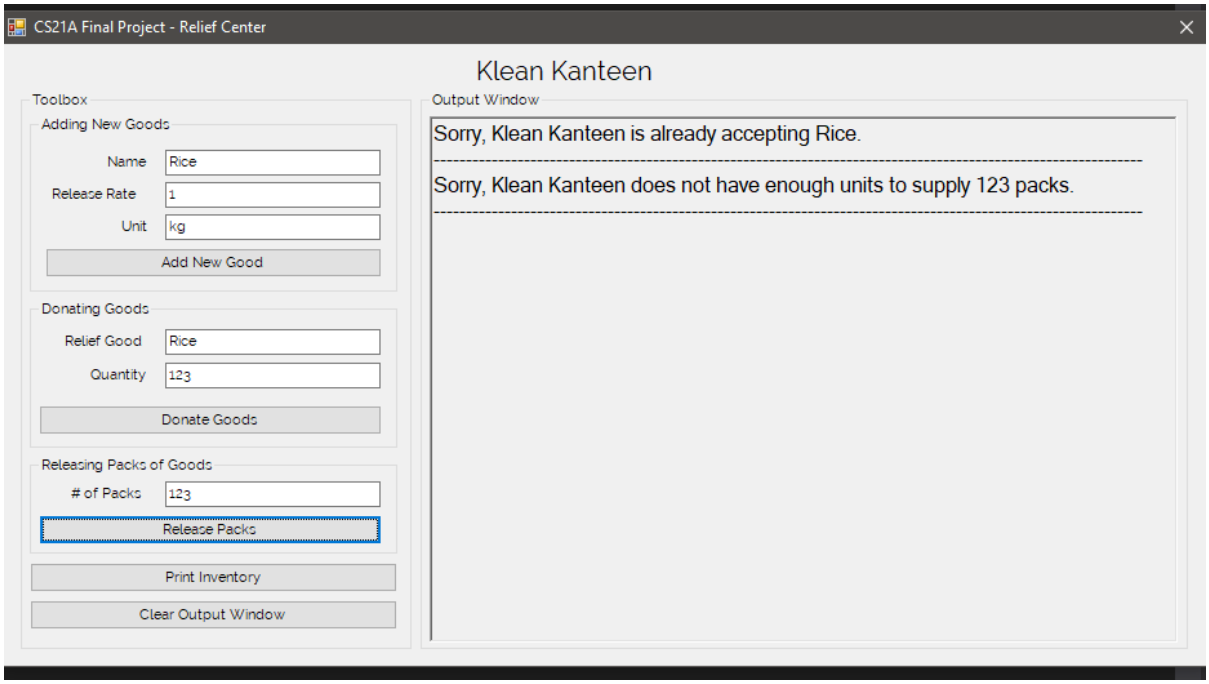
- Include the ability to add custom units. Use a dropdown for units when adding new ReliefGood objects instead of manually typing the unit.
- Use a dropdown for selecting which ReliefGoods to donate instead of typing the name of the ReliefGood.
- Prompt the user for a ReliefCenter name upon startup in a pop-up window. (Hint: use C# Properties to pass data from one form to another)
- Check for invalid input and show a MessageBox with an appropriate message if an error is detected.
    - Empty textbox
    - Invalid numbers (negative, non-numeric strings)
- If RichTextBox is used as an output window, enable auto-scrolling. A button enables clearing the text of the output window.
- Add useful images, icons, and / or sounds.
- Use pop-up windows to prompt users when Adding, Donating, or Releasing goods.
- Handle multiple ReliefCenter objects. Adding and deleting ReliefCenters may be included.
- Deleting or editing accepted goods.

Minimum GUI requirement

Attached are photos of an example GUI aligned with the minimum requirements for this project. Note that "Clear Output Window" is **not required**. With minimum requirements, assume that input will always be valid.






CS21A Final Project - Relief Center
Klean Kanteen
Toolbox
Adding New Goods
Name
Rice
Release Rate
1
Unit
kg
Add New Good
Donating Goods
Relief Good
Rice
Quantity
123
Donate Goods
Releasing Packs of Goods
# of Packs
123
Release Packs
Print Inventory
Clear Output Window
Output Window
Sorry, Klean Kanteen is already accepting Rice.
Sorry, Klean Kanteen does not have enough units to supply 123 packs.