CMSC 203

Spring 2020

Assignment #2

Concepts tested by this program:

**Selection control statements**

**Repetition control statements**

**Input validation loops**

**Relational and logical operators**

**Use a worker class**

**Create a driver class**

**Currency format**

**Random number generation**

**GitHub**

**BIRTHDAY GIFTS**

## Specifications

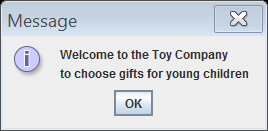
You have decided to use a specialized toy company to purchase birthday gifts for the young children of your friends and relatives. The toys you can choose are divided into three categories: plushies, blocks, and books. You can add a card and/or a balloon with each gift. Your program must display the information of the toy and the cost (including the card and the balloon), and the total amount of the order. Your program must generate a random number between 1-100000 to serve as the order number, followed by your name as programmer. The Toy class you are given will keep a total cost for one toy and determine if the toy is age-appropriate for the child. You should **not make any changes to the Toy class**. Assume that the user enters the name and age of the child correctly.

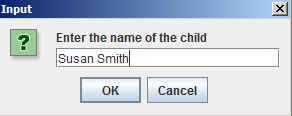
**Note**: Your program should ask for the information of the next toy and repeat the process until the user chooses not to. You DO NOT need to keep the information of toy after displaying it to the user.

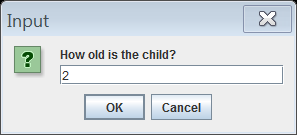
***Requirements***

* All input items and messages will use Dialog boxes. NOTE: if you use a Mac, you must use Alerts and TextInputDialog; if you use a Windows machine, JOptionPane is slightly simpler, or you could use the Alerts/TextInputDialog option. (See Week 2, Module 4)
* Use a Dialog box to display the welcome message.
* Ask for the name of the child.
* Ask for the age of the child.
* Ask for the toy choice and validate the input choice.
* Print out a message if the toy is not age appropriate and ask if the user wants to cancel that toy request.
* If the user replies "yes", repeat the steps for the name, age, and toy choice. If the user replies "no", process the toy requested.
* Ask if a card or balloon should be added to the gift.
* Display the name, age and the total for gift.
* Ask if another gift is desired. If yes, repeat the steps starting with the name of the child.
* If no, display on the console the total amount of the order, a random five-digit order number, and the programmer name.
* Name this driver class Birthday
* Submit the files Birthday.java and Toy.java to your repository in GitHub (see Lab 1)

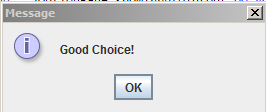
Sample Dialog Boxes

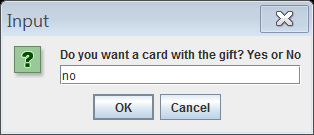


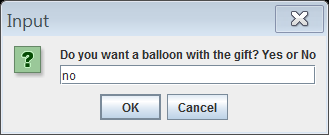


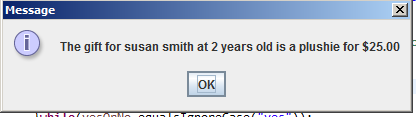


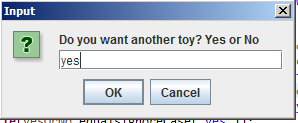


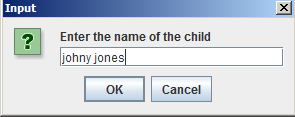


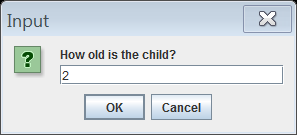


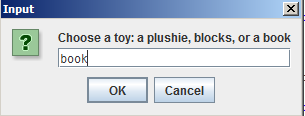




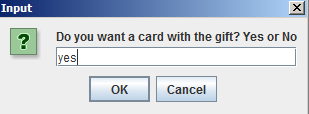


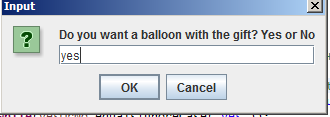


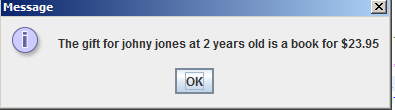


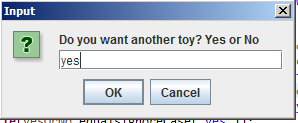


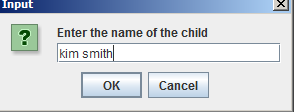


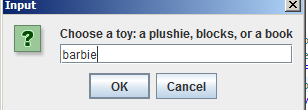


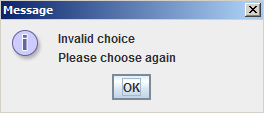


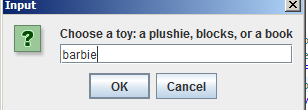












***………. (The Process continues until no more toy is entered)***



***Design***

* Design the application using pseudo-code.
* Create test cases that contain the expected results for each case.
* **One week from today** turn in your initial design on Blackboard.

***Implementation and Documentation:***

* Turn in the following on Blackboard two weeks from today, in a zip file:
* Your driver (Birthday.java) java file and Toy.java (unchanged)
* Two screen shots (from Eclipse’s console **or** a command-line console) of the program executing with two test cases.
* Table of test cases showing 4 test cases (2 with valid toys, 2 showing error messages, DIFFERENT from the below sample)

Sample Test Case

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| NAME | AGE | TOY | MESSAGES | CARD | BALLOON | COST/TOTAL |
| Susie Smith | 2 | plushie | none | no | no | $25.00 |
| Johnny Jones | 5 | blocks | none | no | no | $20.00 |
|  |  |  |  |  |  | $45.00 |

***Good Faith Attempt***

To satisfy the “Good Faith Attempt” (see Blackboard) your code must compile and print the output without finding the name, age, type of toy, cost, or order number. Your GFA need not show the prompts, create the Toy object, or compute the cost or order number. The output for the GFA will be:

The gift for x 0 years old is blocks $0.00

The total cost of your order is $0.00

Order number is 0

Programmer: insert your name here

Note that the GFA is not graded, so you must submit your totally-working code by the submission deadline to have a non-zero grade.

**Grading Rubric**

**CMSC 203 Project #2**

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PROGRAMMING**

Compiles 40 pts \_\_\_\_\_

Accuracy

Passes test cases 15 pts \_\_\_\_\_

Passes private instructor tests 15 pts \_\_\_\_\_

Execution: runs without errors (either run-time or logic errors) 30 pts \_\_\_\_\_

Possible Sub-total 100 pts \_\_\_\_\_

**REQUIREMENTS** (Subtracts from Programming total)

**Documentation:**

Screen shots of two runs were not provided - 4 pts \_\_\_\_\_

Internal class documentation (**within source code**)

Class description, author’s name (with heading comments and

at the end of the code), comments with each section of code - 3 pts \_\_\_\_\_

Screen shot of project in GitHub was not provided - 3 pts \_\_\_\_\_

**Programming Style:**

Incorrect use of indentation, statements, structures - 4 pts \_\_\_\_\_

User interface

Not clear to user how data is to be entered; UI does not follow requirements - 5 pts \_\_\_\_\_

Output is easy to understand - 4 pts \_\_\_\_\_

Dialog boxes and console output conform to specifications - 4 pts \_\_\_\_\_

**Design:**

Implementation does not match design - 5 pts \_\_\_\_\_

Birthday – class used and correctly named - 4 pts \_\_\_\_\_

Prints program title one time on the console at the program beginning

and the welcome message once in a message box at the beginning - 4 pts \_\_\_\_\_

while validation loop for toy choice - 4 pts \_\_\_\_\_

do-while loop to allow for multiple toys to be entered - 4 pts \_\_\_\_\_

Uses Random number generator correctly - 4 pts \_\_\_\_\_

Uses Toy class methods - 4 pts \_\_\_\_\_

Appropriate choices of programming constructs to implement specifications - 4 pts \_\_\_\_\_

(e.g., "continue", “break”, and "return" not used)

Possible decrements: -60 pts \_\_\_\_\_

Possible total grade: 100 pts \_\_\_\_\_