

King Saud University
College of Computer & Information Science
CSC111 – Lab05
Loops
All Sections

Objectives:

The students should learn how to:

1. Follow the loop design strategy to develop loops.
2. Control a loop with a sentinel value.
3. Write loops using for statements.
4. Write nested loops.
5. Combine loops and control statements to solve problems with complex logic.

Submission Instructions:

1. **Due date: Saturday, October 9th, 2021 at 11:59 pm**
2. You can discuss answers with your colleagues. **But no copying.**
3. Submit it to lms.ksu.edu.sa. **Email submissions will not be accepted.**
4. All classes in one java project. The project name must be:
Lab05 ID FirstName LastName.zip. For example:
Lab05_123456789_Marwan_Almaymoni.zip
5. Use the default package.
6. Write your name and university ID as a comment at the start of all java files.

Lab Exercise 1

Write a Java program that calculates and prints the cost of games that a customer buys at a gaming store as following:

- The cost of the game is input.
- A customer must buy at least 1 game (otherwise print "Error").
- If a customer buys more than 2 games, then he will get a 20% discount.

Your program should read game id, the cost of the game as a double value and number of games. Then it should print the total cost after discount (if applicable). Name your class **GameStore1**.

Sample Runs:

```
Welcome to Gaming Center :).
Please, enter game id: 1 ↵
Please, enter the price of a game: 100 ↵
Please, enter the number of games: 5 ↵
Total price for game 1 is: 400.0SR
```

```
Welcome to Gaming Center :).
Please, enter game id: 3 ↵
Please, enter the price of a game: 200 ↵
Please, enter the number of games: 2 ↵
Total price for game 3 is: 400.0SR
```

```
Welcome to Gaming Center :).
Please, enter game id: 6 ↵
Please, enter the price of a game: 200 ↵
Please, enter the number of games: 0 ↵
Error
```

Lab Exercise 2

The previous program has a problem since it does not allow you to enter different prices for different games. Convert your program into an interactive point of sale program for a gaming store. The new program should work as follows:

- The program will read the id and price of games until the user enters -1 as a game id.
- If a customer buys more than 2 copies, then he will get a 20% discount otherwise he will pay the regular price.
- The program should print price before discount, discounted amount, and price after discount.

Name your class **GameStore2**.

Here are some sample runs to show different cases:

```
Welcome to Gaming Center :).
Please, enter game id: 1 ↵
Please, enter the price of next game: 100 ↵
Please, enter game id: 2 ↵
Please, enter the price of next game: 130 ↵
Please, enter game id: 4 ↵
Please, enter the price of next game: 200 ↵
Please, enter game id: -1 ↵
Total price before the discount: 430.0SR
Your discount is: 86.0SR
Total price after discount: 344.0SR
```

```
Welcome to Gaming Center :).
Please, enter game id: 1 ↵
Please, enter the price of next game: 100 ↵
Please, enter game id: 2 ↵
Please, enter the price of next game: 200 ↵
Please, enter game id: -1 ↵
Total price before the discount: 300.0SR
Your discount is: 0.0SR
Total price after discount: 300.0SR
```

Lab Exercise 3

We are going to change the previous program to add even more interactivity and logic to it. The new program should present the user with an options menu that has two options, to add or sell games as shown below:

[illegible]

- If the user chooses “**add**” then he will be able to add new games to inventory. When adding a game, the user needs to provide the game id only. Adding ends when id entered is -1.
- If the user chooses “**sell**” then the program works like the previous one except for one thing. You must make sure that the user cannot sell more games than he added. In other words, allow the user to sell games until there are no more games in inventory.
- All discount rules from the previous program apply here.

(Note: at this stage, you do not have to match ids when selling and adding since you need *arrays* for this.) Name your class **GameStore3**.

(**Note:** unlike other primitive data types like **int** and **double**, to compare two **String** variables **s1** and **s2** use **s1.equals(s2)**. Do NOT use **s1 == s2**)

Here is a sample run to show different cases:

[illegible]

```

*****
Enter your option :> sell ↵
Sorry. There are no more games in store :(
*****
*                               *
*       Welcome to the Gaming Center :)       *
*       -----                               *
* Please enter one of the following options:   *
* 1) add ==> this allows you to add a game to inventory *
* 2) sell ==> this allows you to sell games to a customer *
* 3) exit ==> to end this program             *
*                                              *
*****
Enter your option :> add ↵
Please, enter game id (-1 to end): 1 ↵
Please, enter game id (-1 to end): 2 ↵
Please, enter game id (-1 to end): 3 ↵
Please, enter game id (-1 to end): 4 ↵
Please, enter game id (-1 to end): -1 ↵

*****
*                               *
*       Welcome to the Gaming Center :)       *
*       -----                               *
* Please enter one of the following options:   *
* 1) add ==> this allows you to add a game to inventory *
* 2) sell ==> this allows you to sell games to a customer *
* 3) exit ==> to end this program             *
*                                              *
*****
Enter your option :> sell ↵
Please, enter game id (-1 to end): 1 ↵
Please, enter the price of next game: 100 ↵
Please, enter game id (-1 to end): 10 ↵
Please, enter the price of next game: 200 ↵
Please, enter game id (-1 to end): 6 ↵
Please, enter the price of next game: 120 ↵
Please, enter game id (-1 to end): -1 ↵
Total price before discount: 420.0SR
Your discount is: 84.0SR
Total price after discount: 336.0SR
*****
*                               *
*       Welcome to the Gaming Center :)       *
*       -----                               *
* Please enter one of the following options:   *
* 1) add ==> this allows you to add a game to inventory *
* 2) sell ==> this allows you to sell games to a customer *
* 3) exit ==> to end this program             *
*                                              *
*****
Enter your option :> sell ↵
Please, enter game id (-1 to end): 1 ↵
Please, enter the price of next game: 100 ↵
Cannot sell more games. Out of stock :(
Total price before the discount: 100.0SR

```

```

Your discount is: 0.0SR
Total price after discount: 100.0SR
*****
*           Welcome to the Gaming Center :)           *
*           -----           *
* Please enter one of the following options:           *
* 1) add ==> this allows you to add a game to inventory *
* 2) sell ==> this allows you to sell games to a customer *
* 3) exit ==> to end this program                       *
*                                                         *
*****
Enter your option :> exit ↵
Thanks. Goodbye!

```

Lab Exercise 4

Convert your program into an interactive game-store managing program. The new program should let the user enter data for a new game sale, calculates the revenue and then asks the user if he wants to continue. If the user answers “yes”, the program should keep reading game sales and calculating the revenue. It only terminates when the user answers “no”. Print total revenue for all sales before terminating the program.

(**Note:** unlike other primitive data types like **int** and **double**, to compare two **String** variables **s1** and **s2** use **s1.equals(s2)**. Do NOT use **s1 == s2**

The program asks for the game type ('g' or 'n').

- g : The price of a game should be increased by 40%. For example, if the price was 100, it'll be 140. 20% discount if more than 2 games were bought (first game is not included in the discount).
- n : 10% discount if more than 3 games were bought (first and second games are not included in the discount).
- Anything else will get an error message

Here is a sample run of the program

```

Welcome to Gaming Center :).
Please, enter the type of the game: g ↵
Please, enter the price of a game: 100 ↵

```

```
Please, enter the number of copies: 1 ↵
Total price is: 140.0
Do you want to continue? yes or no: yes ↵
Please, enter the type of the game: g ↵
Please, enter the price of a game: 100 ↵
Please, enter the number of copies: 2 ↵
Total price is: 280.0
Do you want to continue? yes or no: yes ↵
Please, enter the type of the game: g ↵
Please, enter the price of a game: 100 ↵
Please, enter the number of copies: 4 ↵
Total price is: 476.0
Do you want to continue? yes or no: yes ↵
Please, enter the type of the game: n ↵
Please, enter the price of a game: 100 ↵
Please, enter the number of copies: 3 ↵
Total price is: 300.0
Do you want to continue? yes or no: yes ↵
Please, enter the type of the game: n ↵
Please, enter the price of a game: 100 ↵
Please, enter the number of copies: 4 ↵
Total price is: 380.0
Do you want to continue? yes or no: no ↵
The total revenue is 1576.0
```