**HW04 Control Structures**

Due Date: **See Canvas**

Purpose: Learn to use while and for loops. Learn to combine if statements, while and for loops.

Effort: **Individual**: Read [CS Academic Integrity .pdf](https://drive.google.com/file/d/1vHXGQPPRUXZQrUxfnM6hJQid9jio7xlt/view?usp=sharing)

As a CS department we want to make sure we follow due process if we feel a code violation occurred. See [UCCS Student Academic Ethics Code](https://vcaf.uccs.edu/sites/g/files/kjihxj1631/files/inline-files/2017-JUN-27_200-019%20Academic%20Ethics%20Code-APPROVED.docx.pdf). This means all people involved in the violation, whether you took someone else's work or you gave your work to someone else. **These violations can lead to expulsion**. If a Code violation has occurred (due either to my own observation or due to a report by a third party) the following will happen

* Receive a grade of zero for the assignment.
* Have their names placed on the CS department’s list for academic violations.
* Have their names sent to the Dean of Students.

Points: **100 (see rubric in canvas)**

Deliverables: **Upload your design document with your reflection/learning question and your .java file as separate files. Do not upload as a zip file.**

## Assignment Description

Write a program to simulate a self-service bike share station.



The bike share provides the following:

* The bike share station simulation starts with **9 bikes**.
* Rental options: *1-Hour Rental*, *2-Hour Rental*, *6-Hour Rental*, and *Returning a Bike*
* Limits each rental transaction to a maximum of 4 bikes as long as 4 bikes are available in station
* Displays the unlock codes for each rental bike
* Displays a receipt for each rental
* Has a shutdown mode (shutting the bike station down stops the simulation)
* View this website to learn about bike share stations: <https://www.capitalbikeshare.com/>

**Acceptance Criteria**

1. Display a menu that
   1. Shows the number of bikes available
   2. Shows rental options:
      1. 1-hour Pass – set price to **$1.50**
      2. 2-hour Pass – set price to **$2.50**
      3. 6-hour Pass – set price to **$4.00**
      4. Return Bike
2. Prompt customer for:
   1. The rental option
   2. The number of bikes to rent - max allowed is 4
   3. For each prompt, the code **must validate the user input and allow reentry** before moving forward. This means, for each prompt, the code must loop until the user input is valid.
   4. When there is a user input error, ask user to reenter before doing any more processing.
      1. The code must handle user validation for the **menu selection** and **number of bikes to rent**.
      2. If invalid input is entered, repeat the question for the menu selection or number of bikes until a valid value is entered. 

Select rental option 1, 2, 3, or 4: **0**

Invalid entry. Enter 1, 2, 3, or 4: **5**

Invalid entry. Enter 1, 2, 3, or 4: **1**

How many bikes do you want to rent? **Limit is 4:** **0**

Invalid entry. Enter number between 1 and 3: **4**

Invalid entry. Enter number between 1 and 3: **1**

1. If option is *1-Hour Pass, 2-Hour Pass,* or *6-Hour pass* display a receipt that contains:
   1. Rental option selected (1, 2, or 6 hours)
   2. Number of bikes rented
   3. Total rental cost
   4. Unique unlock code for each rental bike
      1. Create a *for loop* that iterates based on the number of bikes.
      2. Generate unique random numbers to represent unlock codes.
      3. Unlock code must be a 5-digit number

Unlock code for bike# 1: 21260

Unlock code for bike# 2: 51043

1. If option is *Return Bike*, display a “bike was successfully returned” statement. Note:
   1. The station must have room, see **Must-Do** section for information on full station
   2. Only one bike at a time can be returned.
2. If option is *shut down* the station
   1. Code must exist the main loop (customer mode) and display grand totals
3. The bike share station simulation starts with **9 bikes**.
   1. Code must keep track of the number of bikes currently available.
   2. Code must handle if station is full and there is no room to return a bike
   3. Code must handle a request where the station does not have enough bikes:
   * Full station: code must indicate there are no empty docks. For example, when first running the code, the station is full so if the user tries to return a bike (option 4 in menu), the code needs to indicate this fact.

Select menu option 1, 2, 3, or 4: **4**

Station is full, please use a different station.

* + Requesting more bikes than are available: code must indicate there are not enough bikes available. In this example, only 2 bikes are available and the user wants to rent 3 bikes for 2-hours:

Select rental option 1, 2, 3, or 4: **2**

How many bikes do you want to rent? The limit is 3: **3**

There are only 2 bikes available.

Request cannot be fulfilled. Please use a different station.

1. Provide a way to **shut down the station** and **print a report** with grand totals for the station.
   1. Essentially, we need a way to stop the main customer loop and end the program. This option is not listed for customers to see.
   2. We’ll call this shutting down the bike station.
   3. Perform these steps:
      1. At the main menu, when asked for the rental option, an employee enters the shutdown code **999**.
      2. If the shutdown code **999** is entered, then the while loop (customer mode) must end. This means, the code will exit the loop and create a report showing:
         1. Total number of bikes rented for **all customers**
         2. Total sales for **all customers**
2. See sample output below.

## Requirements Analysis and Design

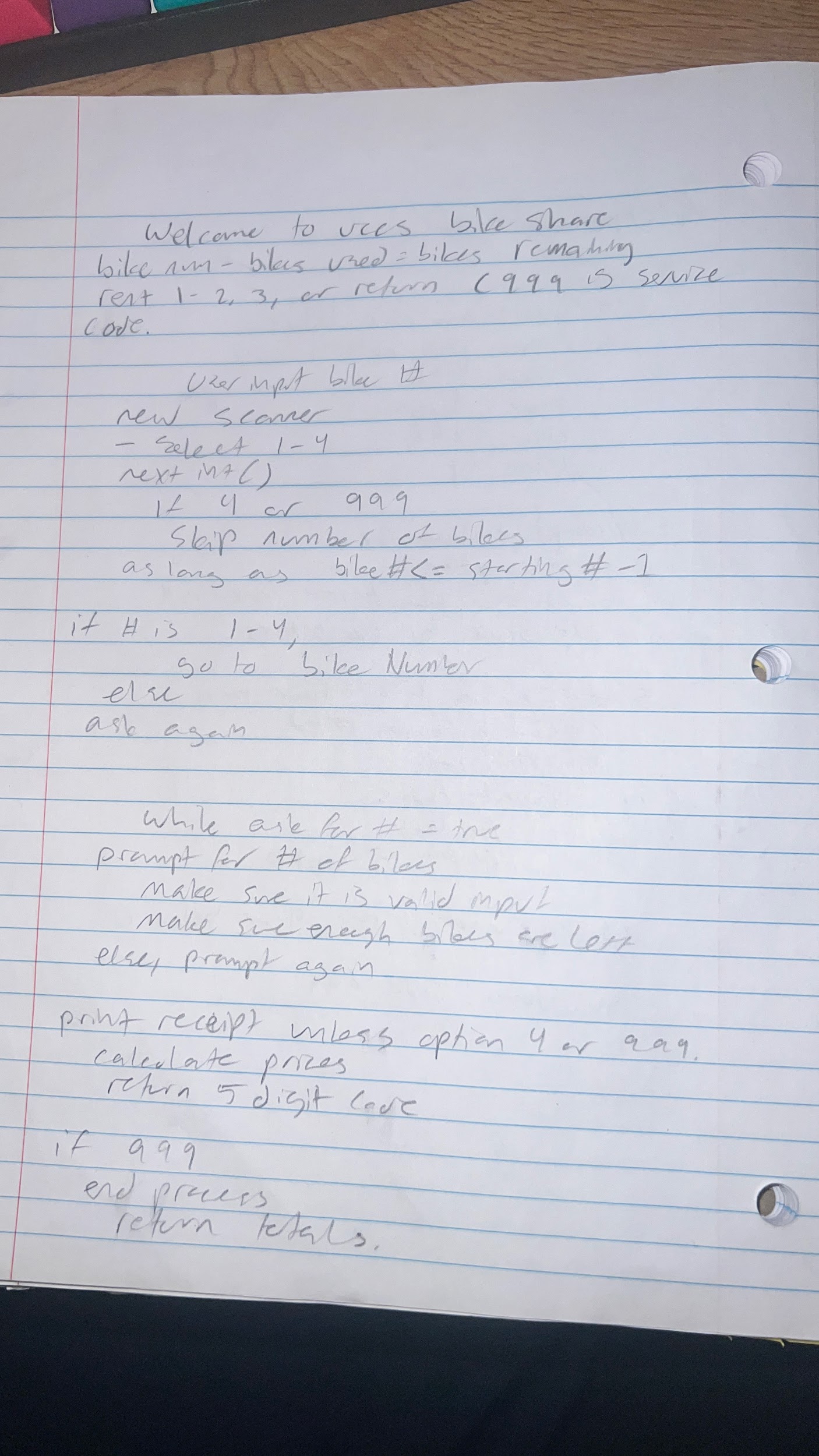
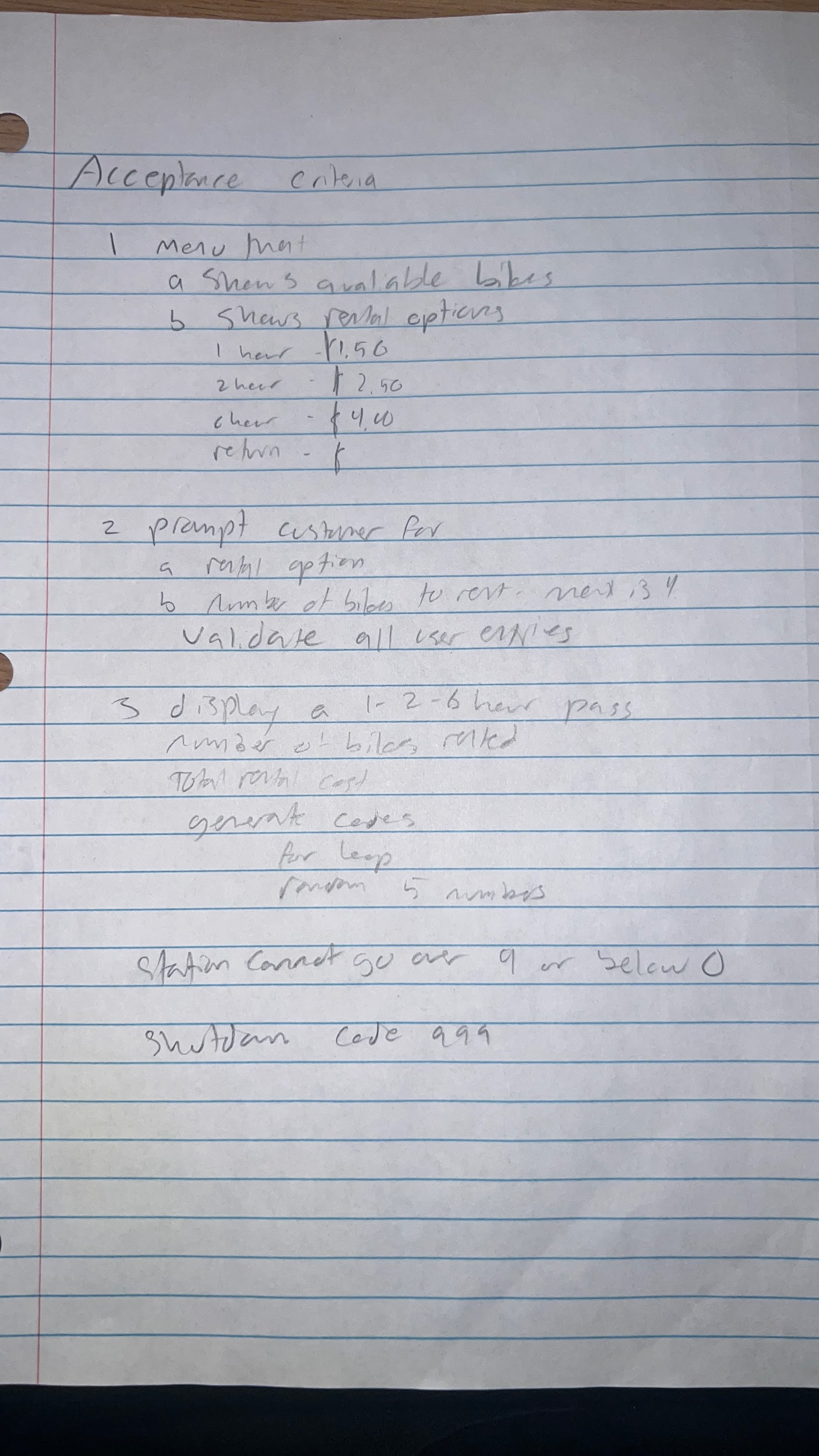
Read slides 1-11 on  [Problem Solving and the Software Development Life Cycle](https://docs.google.com/presentation/d/e/2PACX-1vSC4tM0BsOaYjRmd_emNeHtKrWaExKXfvvLxQC10rCt77CPT_WvF_s3W--o_yuj8XczYGo7qHxgjMIq/pub?start=false&loop=false&delayms=3000&slide=id.p1) and review the last homework examples of breaking into tasks, creating test cases and writing pseudocode.

Think about where you will use if/else or switch statements. Do not write a code solution until you complete this first. This is to get ideas on paper and practice problem solving. Create a document and include the following in your design.

**Requirements Analysis - Tasks**

* Break the problem into smaller pieces by listing all tasks needed to implement a solution.
* Put in the design document or on paper.

**Requirements Analysis - Test Cases**

* A table for each task needed to solve the problem with a wide range of possible inputs and outputs to help ensure functional correctness.
* This can be on paper or in the design document.
* 

**Resources**

* List at least 4 resources that are or will be useful for the assignment. This should be in the document so it can include links.
* Resource Location -can be any of the following -
  + Lecture Number and slide(s) -
  + Section from the book -
  + JavaTPoint : <https://www.javatpoint.com/java-tutorial>

1- L09CH04 Slide 9 Flow of control

2-<https://www.javatpoint.com/java-do-while-loop> - do loop syntax

3- my HO03- making random numbers

4-<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html> - Switch Statements

**Design Possible Solution**

* Hand written - paper, whiteboard or tablet. This might include a sketch of ideas to understand the problem.
* For each task include a pseudocode of steps (algorithm). Put images from your writing in the final design document.

## Specification for Implementing Code

[Review Problem Solving Using Software Development Life Cycle](https://docs.google.com/presentation/d/e/2PACX-1vSC4tM0BsOaYjRmd_emNeHtKrWaExKXfvvLxQC10rCt77CPT_WvF_s3W--o_yuj8XczYGo7qHxgjMIq/pub?start=false&loop=false&delayms=3000&slide=id.ge6997ea817_0_125) slides 12 - 19

1. Create a Java class **LastNameFirstNameHW03** within that project calledwithin the **CS1150HW** project
2. Follow [Problem Solving and Software Development Life Cycle](https://docs.google.com/presentation/d/e/2PACX-1vSC4tM0BsOaYjRmd_emNeHtKrWaExKXfvvLxQC10rCt77CPT_WvF_s3W--o_yuj8XczYGo7qHxgjMIq/pub?start=false&loop=false&delayms=3000&slide=id.p1)
3. Remember to
   1. use **correct data types** and **constants**
   2. Use proper if/else conditions
   3. Uses a **switch statement** to display the number of days
4. Main code structure
   1. Must use a *while loop* that waits for customer input.
   2. Must use a *for loop* to create and display the unlock code for each bike being rented.

**boolean** stationInService = **true**;

**while** (stationInService) {

// Inside loop you will change stationInService to false

// when 999 is entered

}

1. Use the ***random*** method in the Math class – **Math.*random*()** - see book and lecture
2. Use *toUpperCase* or *toLowerCase* method in the Character class
   1. To write simpler and cleaner code that performs *user validation,* convert the theme park character to an uppercase value or lowercase value using **Character** class
3. Write code that:
   1. **DO NOT** use System.exit(0) to exit program if an error occurs
   2. **DO NOT** use break or return statements if an error occurs
   3. Using System.exit, return, or break(break can only be used for switch cases) will result in loss of points for correctness
   4. The purpose is to learn to write properly nested if-statements.
4. **Write the code incrementally**
   1. Focus your thinking on one task at a time, write that code, test and then go to the next task.
   2. Write the skeleton while loop that displays the menu and reads the number of bikes desired.
   3. Next add code to determine if in customer mode or shut down mode.
   4. Test that the loop ends when shut down code (999) is entered.
   5. Keep adding in one new task of functionality and get it working before adding in anything else.

**Output**

Your output should look similar to the following:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to UCCS Bike Share

9 bikes are available Note indication of # bikes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Rental Options Fee

-------------------------------------------------

1) 1-Hour Pass $1.50

2) 2-Hour Pass $2.50

3) 6-Hour Pass $4.00

4) Return Bike

-------------------------------------------------

Values entered by user

Select rental option 1, 2, 3, or 4: **1**

How many bikes do you want to rent? The limit is 4: **4**

-------------------------------------

-------------- Receipt --------------

-------------------------------------

1-hour rental for 4 bikes

Unlock code for bike# 1: 60725

Unlock code for bike# 2: 94169

Unlock code for bike# 3: 43749

Unlock code for bike# 4: 71823

Rental Cost: $6.00

Thank you for your business!

-------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome TO UCCS Bike Share Note that there are now only

5 bikes are available 5 bikes available.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Rental Options Fee

-------------------------------------------------

1) 1-Hour Pass $1.50

2) 2-Hour Pass $2.50

3) 6-Hour Pass $4.00

4) Return Bike

-------------------------------------------------

Select menu option 1, 2, 3, or 4: **0** Example of handling

Invalid entry. Enter 1, 2, 3, or 4: **5** invalid menu option

Invalid entry. Enter 1, 2, 3, or 4: **2**

How many bikes do you want to rent? The limit is 4: **0** And invalid # bikes

Invalid entry. Enter number between 1 and 5: **5**

Invalid entry. Enter number between 1 and 5: **3**

-------------------------------------

-------------- Receipt --------------

-------------------------------------

2-hour rental for 3 bikes

Unlock code for bike# 1: 77256

Unlock code for bike# 2: 60947

Unlock code for bike# 3: 33314

Rental Cost: $7.50

Thank you for your business!

-------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome to UCCS Bike Share Station is down to 2 bikes

2 bikes are available

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Rental Options Fee

-------------------------------------------------

1) 1-Hour Pass $1.50

2) 2-Hour Pass $2.50

3) 6-Hour Pass $4.00

4) Return Bike

-------------------------------------------------

Case when the # of

Select rental option 1, 2, 3, or 4: **1** bikes available can

How many bikes do you want to rent? The limit is 4: **3** not fulfil request

There are only 2 bikes available.

Request cannot be fulfilled. Please use a different station.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome TO UCCS Bike Share

2 bikes are available

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Rental Options Fee

-------------------------------------------------

1) 1-Hour Pass $1.50

2) 2-Hour Pass $2.50

3) 6-Hour Pass $4.00

4) Return Bike

------------------------------------------------- Successfully returning a bike 

Select menu option 1, 2, 3, or 4: **4** Now station has 3 bikes

Bike was successfully returned

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome TO UCCS Bike Share

3 bikes are available

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Rental Options Fee

-------------------------------------------------

1) 1-Hour Pass $1.50

2) 2-Hour Pass $2.50

3) 6-Hour Pass $4.00

4) Return Bike

-------------------------------------------------

Example showing station shut down using **999** for menu option

Select menu option 1, 2, 3, or 4: **999**

UCCS Bike Station at Main Hall was successfully shut down

Total Bikes Rented = 7 **total number of bikes** rented 

Total Sales = $13.50 and **total sales** for all customers