

## **Overview of Project**

The final project is to create a program to shop for activities. The program set up a catalog of activities and then allow the user to browse and purchase activities from the catalog (database). The project will incorporate an API, JSON parsing, a database, Object-Oriented Programming, and an interactive user interface.

API: <https://www.boredapi.com/>

Database: SQLite

## **Coding Standards**

The final project code is expected to follow the coding standards described below. Examples are provided in the “Coding\_Standards.pptx”. Points will be deducted if any of the following standards are not met.

- Variables and functions should have meaningful names
- Code should have comments and be easy to read
- Avoid hard-coding
- Remove any unused code
- Add descriptions when you display output to the screen
- Every function must have a doc string
  - Use triple quotes
  - Describe the goal of the function
  - Each input needs name, data type, and description
  - Each output needs name and data type

## **Project Part 1**

The first part of the project includes setting up the catalog of activities and allowing the user to browse the activities.

The activities will be collected from the <https://www.boredapi.com/>. The documentation for using the API can be found on the website. The activities collected from the API will be in JSON format which will be parsed to and used to create a database of activities (catalog).

Once the database has been created, write code to allow a user to browse the activities in the database by filtering by one or more attributes of the activities (i.e., type, price etc.) and displaying the filtered set of activities. To allow the user to interact with the program, create an interactive user interface to browse the database of activities. This user interface will be expanded upon in Part 2 of the project.

### **RUBRIC**

Objective	Description	
Get activities	Get random activities from the API by submitting GET request(s) to the API	
Create database of activities	Create an SQLite Database of <b>150</b> unique random activities	
Browse activities	Allow the user to browse the activities in the database by one or more of the attributes of an activity	
User Interface	Create an interactive user interface to browse the database of activities	
Run the program	The program needs to have a "main.py" file that can be run to start the program.	
Coding standards	All coding standards are met in all files and functions.	
No errors	Code runs without errors.	
Zip file	Code is submitted in a .zip file called "YOUR_NAME_Project_Part1.zip"	

## **Project Part 2**

The second part of the project includes creating a customer object with a cart. The customer should be able to add activities to their cart, view their cart with activities and their associated price, and purchase items in their cart.

The customer class needs to have an attribute to represent a “cart”. The cart needs to contain information about the activities added to it including the price of each activity. The customer needs to be able to view their cart with the activities and total price. The customer needs to be able to “purchase” the items in the cart which will empty the cart.

### **RUBRIC**

<b>Objective</b>	<b>Description</b>	
Customer class	Create a customer class with a “cart”	
Create a new customer	Create a customer object	
Customer can view cart	The customer (user) can view the cart contents and total price of all activities	
Customer can add an activity	The customer can add an activity to the cart	
Customer can purchase items in the cart	The customer can purchase the items in the cart and the cart is empty after the purchase	
Coding standards	All coding standards are met in all files and functions.	
No errors	Code runs without errors.	
Zip file	All project code is submitted in a .zip file called “YOUR_NAME_Project_Part2.zip” (this should also contain the code from Part 1)	