**02.Snow Sport Store**



Write a class **SnowSportStore**, which implements the following functionality:

**Functionality**

**Constructor**

Should have these **3** properties:

* **storeName –** A string representing the name of the store.
* **availableEquipment –** An array to store the available sports equipment in the store.
* **revenue –** A number initialized to **0**, representing the store's total earnings.

**At the initialization of the SnowSportStore** class**,** the **constructor** accepts the **storeName.** The **revenue** has a **default value of 0,** and the **availableEquipment** array is empty.

**Hint:** You can add more properties to help you finish the task.

**addEquipment(type, price, condition) -** This method **adds** new equipment to the store. It accepts the following **arguments**:

* If any of the following requirements is **NOT fulfilled**, an **error** with the following message should be **thrown**:

**"Invalid equipment details!"**

* + **type** – non-empty string;
  + **price** – positive integer number;
  + **condition** – non-empty string;
* Otherwise, the equipment is added to the **availableEquipment** array as an object with the properties **{type, price, condition}**, and the method **returns**:

**"New equipment added: {type} / {condition} condition - {price}$."**

* When **returning** the result, the **Price** mustbe **rounded to the second decimal point!**

**rentEquipment(type, rentalDays) –** This method rents out equipment. It accepts the following arguments:

* + **type** – non-empty string;
  + **rentalDays** – positive integer representing the number of days the equipment is rented for;

**Note:** No additional validation for the parameters is required.

* The method searches for equipment in the **availableEquipment** array where the type **matches** and the condition is **used**.
* If no matching equipment is found, an **error is thrown** with the message:

**"{type} is not available for rent!"**

* Otherwise, the rental price is calculated as **10%** of the equipment **price per day**, **multiplied** by the number of rental **days**:

**rentalCost = price \* 0.1 \* rentalDays**

* Finally, you must add the **soldPrice** to the **revenue** and return:

**"{type} rented for {rentalDays} days. Total cost: {rentalCost}$."**

**Note: rentalCost** must be **rounded** to the second decimal point!

**sellEquipment(type)** - This method sells equipment from the store. It accepts the following argument:

* + **type** – non-empty string representing the type of equipment to sell;
* The method searches for equipment in the **availableEquipment** array where the **type** matches and the condition is **new**.
* If no matching equipment is found, **an error is thrown** with the message:

**"{type} is not available for purchase!"**

* Otherwise, the equipment is removed from the **availableEquipment** array, its price is added to the **revenue**, and the method returns:

**"{type} has been sold for {price}$."**

**Note: price** must be **rounded** to the second decimal point!

**showRevenue()** – This method displays the store's total revenue.

* If the **revenue** is 0, it **returns**:

**"** **Nothing has been sold or rented."**

* Otherwise, it **returns**:

**"{storeName} has made a total revenue of {revenue}$."**

**Note: revenue** must be rounded to the second decimal point!

**Example**

|  |
| --- |
| **Input 1** |
| let store = new SnowSportStore('Alpine Gear Shop');  console.log(store.addEquipment('Ski', 500, 'new'));  console.log(store.addEquipment('Snowboard', 300, 'used'));  console.log(store.addEquipment('Helmet', 50, '')); |

|  |
| --- |
| **Output 1** |
| New equipment added: Ski / new condition - 500.00$.  New equipment added: Snowboard / used condition - 300.00$.  Uncaught Error Error: Invalid equipment details! |

|  |
| --- |
| **Input 2** |
| let store = new SnowSportStore('Alpine Gear Shop');  console.log(store.addEquipment('Ski', 500, 'new'));  console.log(store.addEquipment('Snowboard', 300, 'used'));  console.log(store.rentEquipment('Snowboard', 3));  console.log(store.rentEquipment('Boots', 3)); |

|  |
| --- |
| **Output 2** |
| New equipment added: Ski / new condition - 500.00$.  New equipment added: Snowboard / used condition - 300.00$.  Snowboard rented for 3 days. Total cost: 90.00$.  Uncaught Error Error: Boots is not available for rent! |

|  |
| --- |
| **Input 3** |
| let store = new SnowSportStore('Alpine Gear Shop');  console.log(store.addEquipment('Ski', 500, 'new'));  console.log(store.addEquipment('Snowboard', 300, 'used'));  console.log(store.sellEquipment('Ski'));  console.log(store.sellEquipment('Helmet')); |

|  |
| --- |
| **Output 3** |
| New equipment added: Ski / new condition - 500.00$.  New equipment added: Snowboard / used condition - 300.00$.  Ski has been sold for 500.00$.  Uncaught Error Error: Helmet is not available for purchase! |

|  |
| --- |
| **Input 4** |
| let store = new SnowSportStore('Alpine Gear Shop');  console.log(store.addEquipment('Ski', 500, 'new'));  console.log(store.addEquipment('Boots', 100, 'used'));  console.log(store.addEquipment('Helmet', 200, 'new'));  console.log(store.addEquipment('Snowboard', 300, 'used'));  console.log(store.sellEquipment('Ski'));  console.log(store.sellEquipment('Helmet'));  console.log(store.rentEquipment('Snowboard', 3));  console.log(store.showRevenue()); |

|  |
| --- |
| **Output 4** |
| New equipment added: Ski / new condition - 500.00$.  New equipment added: Boots / used condition - 100.00$.  New equipment added: Helmet / new condition - 200.00$.  New equipment added: Snowboard / used condition - 300.00$.  Ski has been sold for 500.00$.  Helmet has been sold for 200.00$.  Snowboard rented for 3 days. Total cost: 90.00$.  Alpine Gear Shop has made a total revenue of 790.00$. |