Create a new project folder named **vttp5\_sdf\_day08**

Create a project folder named **project01am** in the project folder **vttp5\_sdf\_day08**

Link up your project in the root folder **vttp5\_sdf\_day08** to your git repo named **vttp5\_sdf\_day08**

**Task 1**

Create a **Product.java** entity file in the package **day08**.

Entity: Product

Attributes:

Long id

String prodName

String prodDesc

String prodCat

Float price;

Date createdDate

Generate the Constructors, getter/setters and toString functions.

**Task 2**

In the project folder **project01am**, Create a **App.java** file that contains the java main entry point public void main(String[] args)

**Task 3**

Add the following products into an Product ArrayList in App.java main().

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Category** | **Price** |
| Mouse | For click UI on screen | Computer | 99.0 |
| Keyboard | device that allows alpha numerics inputs | Computer | 235.5 |
| 15.6 inch monitor | Extended display panel | Computer | 157.5 |
| Huawei Pura 70 Ultra | Huawei phone | Mobile | 900.0 |
| Huawei Mate 50 Pr | Huawei phone | Mobile | 1200.0 |
| iPhone 16 Pro | Iphone | Mobile | 2000.0 |
| iPhone 14 Pro | Iphone | Mobile | 1800.0 |

**Task 4**

Filter the list using stream and lambda functions to display mobile phones that are above $1500. Use a for Loop to display the filtered product.

**Task 5**

Accept an argument input that specify a directory and file.

Write the filtered output in Task 4 to the file.

**Task 6**

Given 4 digits or 4 alphabets through console input.

Find the number of permutations.

Put the permutation into Set Collection.

Print out the Set collection.­

Example : ABCD should produce

ABCD 🡪 ACDB, ADBC, ~~ABCD~~, ABDC, ~~ABCD~~

BCDA 🡪 BDAC, BACD, ~~BCDA~~, BDCA, ~~BDAC~~

CDAB 🡪 CABD, CBDA, ~~CDAB~~, CDBA, ~~CDAB~~

DABC 🡪 DBCA, DCAB, ~~DABC~~, DBAC, ~~DBCA~~

**Task 7**

Create a mastermind game on Java Terminal console.

Steps to construct Mastermind game

1. Use random function to generate 4 digits number between 1111 and 6666.
   1. 4 random number generator to generate each digit (1 – 6).
2. Use console to prompt for 4 digits input.
   1. This should be inside a while loop.
3. Check for invalid input (e.g. input not between 1111 and 6666).
4. Use a for loop for check
   1. Check the digit at each index using atChar(index) match position & number.
      1. Increment CP by one if match number & position.
   2. Check whether the input digit occurs in other positions
      1. If point a fulfills, no need to perform this point b.
      2. Loop through to check if there are matching number.
         1. Increment C if number matches but position doesn’t match.
5. Repeat 4 until solution matches.
   1. Allow up to 12 tries
6. Display whetehr you win or lose.
7. Repeat the game.

**Task 8**

Create a websocket server application and client application that performs the following.

Client send a string, i.e. 65 \* 65, to the websocket server.

Server detects the arithmatical operation.

Server computes the arithmatical results.

Server sends the arithmatical results as output to the websocket client.

Both client and server terminates when client send the string ‘end’.