جامعة الحسين التقنية
Al Hussein Technical University

ASSIGNMENT BRIEF

HTU Course No:	30201100	HTU Course Name:	Fundamentals of Computing
BTEC UNIT No:		BTEC UNIT Name:	

Version: 1

جامعة الحسين التقنية
Al Hussein Technical University

Assignment Brief

Student Name/ID Number/Section	
HTU Course Number and Title	**30201100: Fundamentals of Computing**
BTEC Unit Number and Title	**Not Available**
Academic Year	Spring 2021/2022
Assignment Author	Course Instructors
Course Tutor	Eng. Ashraf Alsmadi, Dr. Eyad Taqieddin, Eng. Malek Al-Louzi , Dr. Rami Al-Ouran, Eng. Umar AlKafaween
Assignment Title	Store Management App
Assignment Ref No	**Assignment 1**
Issue Date	17th May 2022
Formative Assessment dates	29th May 2022 to 9th June 2022
Submission Date	18th June 2022
IV Name & Date	Eng. Dania Alsa'id and Eng. Hebah AlDahoud

Submission Format

This assignment is a project design. You must complete it and submit it according to the following guidelines (Failing to follow these guidelines may result in an 'Unclassified' grade):

Project Design Guidelines

- You are required to submit 2 things:
  1) A PDF file that includes:
     - **a)** **The flowchart** you used to implement Task 3, and the **block diagram** with your justifications for Task 8. **(Auto generated flowcharts are not allowed).**
     - **b)** The **student declaration form** attached to this assignment instructions (you may use an electronic signature to sign it).
  2) Your code. Rename code.c to (*YourFullName_FOC.c*) and submit it. Your code should be **fully commented**. All your work of coding and compilation should be done using **Ubuntu and gcc** compiler only.

- Only soft-copy submissions are allowed. You are required to upload your submission files to the university's eLearning system through (https://elearning.htu.edu.jo) within the submission date and time stated above. **NO SUBMISSIONS by EMAIL or TEAMS and NO LATE SUBMISSIONS WILL BE ACCEPTED.** Failing to follow this guideline may result in an 'Unclassified' grade.
- **If you commit any kind of plagiarism, HTU policies and regulations will be applied.**
- This is a strictly individual assignment and no collaboration amongst students is allowed. Working with your colleagues is not considered teamwork, but plagiarism.

In-Class Oral Discussion

- In-class assessment will be one to one online or in-person oral discussion between you and your instructor. It includes designing, developing, debugging, analyzing, and evaluating the code and algorithms developed in this project.
- The attendance of the oral assessment is mandatory in the date and time determined by your instructor. Be ready to **open your camera** throughout the assessment in the case of online discussion.
- You are required to be able to answer any questions about the tasks during the oral assessment. Any questions not answered in the oral assessment with the required level of details means that you might lose its criteria, even if you have already answered it correctly in your submission.
- You must sign the witness form that your instructor will fill up during the discussion to complete the oral assessment process.

جامعة الحسين التقنية
Al Hussein Technical University

Unit Learning Outcomes
LO1 Discuss the basic concepts of computer hardware, software and operating systems, with the basic relation between them, taking C commands as an example. **LO2** Implement a full program in C with identification of the three kinds of control structures: sequence, selection, and repetition, understanding the capabilities of implementing C codes that capable to deal with functions. **LO3** Understanding the basic concepts of pointers in C with implementation of arrays and dealing with files. In addition to the role of C programming as a powerful tool to communicate between devices and manipulate their data.

Assignment Brief and Guidance

The most famous store in Digital Land is called ForMart. They are working on developing a new system for the store, and they asked for your help specifically!

This assignment is broken down into 8 tasks that you need to complete in order to help ForMart continue being the most famous store in all of Digital Land.

Assignment Files

This assignment contains 4 files that you will be using:

- **code.c**, which contains the starting code for the assignment. You will need to implement functions in this file to complete the tasks.
- **products.txt**, which contains the products of the store and their prices.
- **foc_sp22.h**, the header file of the library that allows us to draw pictures from 2D arrays. This file needs to be included inside code.c.
- **foc_sp22.o**, the object file of the library that allows us to draw pictures from 2D arrays. This file needs to be compiled with code.c.

Getting Started

Start by looking at products.txt and code.c.

products.txt

This file has a very specific format. It starts with an integer that represents the number of items in the file. Each line after that has 2 values and represents a product in the store. The first value is an integer, and it is the SKU of the product (like an ID or a barcode), and the second value is a double that represents the price of the product.

code.c

This C file contains the following starting code. It has 6 essential functions, each of them will be completely explained in one of the tasks below, but here is a quick summary:

mainMenu: Prints the main menu and invokes all the other essential functions.

findNearestShop: Finds the nearest shop to the store.

listProductsCheaperThan: Lists all products from products.txt that are cheaper than some price.

addProduct: Adds a new product to products.txt.

processPurchase: Processes a purchase for a customer and creates a receipt.

drawPrices: Draws a figure that shows the prices of some products.

The file also contains the following helper functions, these are simpler functions that you **need** to implement in order to use them in other functions and make your job easier. **It is recommended that you start with implementing these functions before the essential functions.**

manhattanDistance: Computes the Manhattan distance between 2 points. See Task 1 for more information about Manhattan distance.

euclideanDistance: Computes the Euclidean distance between 2 points. See Task 1 for more information about Euclidean distance.

isOnDiscount: Decides whether a product is on discount or not.

discountedPrice: Computes the discounted price of a product. Discounts are always 15%.

getPriceBySku: Gets the price of a product from products.txt.

doublesAreEqual: Compares 2 double values and decides if they are equal or not. This function is already implemented for you, **you do not need to implement it**. Use this function whenever you need to compare 2 double values for equality because it is more accurate than the = = operator.

Note 1: All the functions have comments above them in code.c to remind you of what they do.
Note 2: **Do NOT** modify the names, return types, or arguments of the provided functions.
Note 3: You are allowed to create additional functions if you need to, but you **HAVE TO** implement all the provided functions.

The file also contains some constants that are defined at the beginning of the file. Constants are like variables, but their values cannot be changed. You can use these constants anywhere in your code. Check the comments in the file to understand what each constant is.

The Tasks

Task One

Implement the function int mainMenu().

This function is called from the main function. It should print out the main menu of the program, it looks like figure 1 below.

```
Welcome! Please choose one of the following options:
1. Find Nearest Shop.
2. List products cheaper than a price.
3. Add a new product to the store.
4. Process a Purchase.
5. Draw a figure of the prices.
6. Exit.
```

Figure 1

It then accepts the user's choice, which is an integer from 1 to 6. Depending on the choice of the user, it should call the appropriate function and pass any values it needs to it. The following choices should run the following functions and do the following actions before and after running the functions.

Choice	Function To Call	Before Calling It	After Calling It
1	double findNearestShop()	Do nothing	Print the value returned from the function
2	void listProductsCheaperThan(double price)	Ask the user to input a price and pass it to the function as an argument	Do nothing
3	int addProduct(int sku, double price)	Ask the user to input an SKU and a price for the new product and pass them to the function as arguments	If the product was added successfully, print a success message. If the product was not added successfully, print an error message

4	void processPurchase(int numOfItems, int itemSkus[])	Ask the user to input the number of items the user is going to buy, and then the SKUs of these items. Pass the number of items, and the array of SKUs to the function as arguments	Do nothing
5	void drawPrices(int numOfProducts, int productSkus[])	Ask the user to input the number of products they want to draw the prices for, and then the SKUs of these products. Pass the number of products, and the array of SKUs to the function as arguments	Do nothing

After each choice from 1 to 5, mainMenu should return 0 to the main function. If the user chooses 6 (Exit) the function should return 1 to the main function so it could exit. Any other choice should print "Invalid Choice" and return 0. Check the implementation for the main function if you need to.

Task Two

Implement the function void findNearestShop().

This function asks the user for the coordinates of shops around the store to find the closest shop to restock from.

The store is at location: (SHOP_LOCATION_X, SHOP_LOCATION_Y)
where SHOP_LOCATION_X is the x coordinate of the shop, and SHOP_LOCATION_Y is the y coordinate of the shop. Both values are defined as constants at the top of the program, and you can use them inside this function directly.

To find the distance between 2 points (x_1, y_1) and (x_2, y_2), there are many ways. We know the Euclidean Distance, which is calculated as:

$$\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$$

There is also another way called the Manhattan Distance, which is calculated as:

$$|y_2 - y_1| + |x_2 - x_1|$$

Before you implement `findNearestShop`, it is recommended that you implement the 2 functions `euclideanDistance`, `manhattanDistance` to return the distance between 2 points each depending on its equation above, so you could use them in this task.

When `findNearestShop` starts, it should ask the user to enter a word to decide which distance measure to use (check figure 2):

"euclidean" for Euclidean Distance

"manhattan" for Manhattan Distance

If the user enters any other word, print an error message and ask for the word again.

```
1
Enter the distance measure you would like to use:
Type euclidean for Euclidean distance.
Or type manhattan for Manhattan distance.
```

Figure 2

After the user enters the word, it asks the user to enter the number of shops nearby.

For each shop nearby, it asks the user to enter the coordinates x y of that shop, then computes the distance of that shop to the store using the selected distance measure.

The function should return the distance of the closest shop to the store, which will then be **printed inside the** `mainMenu` function as explained in Task 1.

Task Three

Design a flowchart for the function in **Task Two**. You can draw the flowchart on paper and take pictures of it, but you need to be extra careful to make sure the drawing and the pictures are clear. Alternatively, you can use any programs or free websites to build the flowchart using basic shapes (e.g., draw.io or app.diagrams.net).

Task Four

Implement the function void listProductsCheaperThan(double price).

This function accepts a double value that represents a price and prints on the screen all the products inside products.txt that are cheaper than the provided price. Check figure 3 for an example of what this function prints.

```
2
Please enter a price: 2
Product 64967 has price 0.50.
Product 31402 has price 1.20.
Product 27638 has price 1.40.
Product 42377 has price 0.30.
Product 49250 has price 0.50.
Product 72646 has price 0.85.
Product 14371 has price 0.35.
Product 39044 has price 1.53.
Product 44763 has price 1.20.
Product 66958 has price 1.87.
Product 33439 has price 0.50.
Product 37462 has price 0.34.
```

Figure 3

Some products are on discount. The constant array DISCOUNTED that is defined at the top of the program contains the SKUs of 7 products that are on discount. The discount is always 15%, but the prices in products.txt are before discount. You need to always make sure to use the discounted price if a product is on discount. For example, product 27638 is on discount, its original price is 1.65, but after applying a 15% discount it becomes 1.40.

Before you implement listProductsCheaperThan, it is recommended that you implement the 2 functions isOnDiscount, and discountedPrice, so you could use them in this task.

isOnDiscount: Accepts the SKU of a product and returns 1 if the product is inside the DISCOUNTED array, or 0 otherwise.

discountedPrice: Accepts a price and returns the price after applying a 15% discount.

Task Five

Implement the function `int addProduct(int sku, double price)`.

This function accepts 2 arguments, an SKU and a price for a new product to be added to products.txt. The provided SKU must be unique. If a product already exists with the same SKU, do not add the new product, and the function should return 0. Otherwise, if the product is unique and was added successfully, return 1.

Depending on the returned value, the `mainMenu` function should print an error message or a success message as explained in Task 1.

Note: Remember to preserve the layout of products.txt. This means that after this function finishes, products.txt should have the new number of products on top, then at each line after, there will be 2 values, an SKU and a price of a product. The new product will be added at the bottom of the file.

Task Six

Implement the function `void processPurchase (int numOfItems, int itemSkus[])`.

This function accepts the number of items the customer is buying, and an array of SKUs of the items the customer is buying. The array can contain the same SKU more than one time. The function prints the purchase details on the screen **and also writes them into a file called "receipt.txt"**. Each product is printed on a separate line with the following details: SKU, quantity, and total product price (quantity x price).
The total cost of the whole purchase is printed at the end. Check figure 4.

```
4
Please enter the number of items you are purchasing: 4
Please enter the sku of the item: 27638
Please enter the sku of the item: 27638
Please enter the sku of the item: 72327
Please enter the sku of the item: 37462
Customer purchased 3 items:
- 27638 2 2.80
- 72327 1 2.05
- 37462 1 0.34
===============
Total: 5.19
```

Figure 4

Before you implement `processPurchase`, it is recommended that you implement the function `getPriceBySku`.

`getPriceBySku`: Accepts the SKU of a product and returns the price of that product. If the product is on discount, it returns the discounted price. If the SKU does not belong to any product, it returns -1.

When you implement `processPurchase`, take into account the following notes:

Note 1: If the price is on discount, use the discounted price.
Note 2: If any provided SKU is not in the list of products, PRINT an error message as follows: "SKU 58312 is not available! Ignoring!", where 58312 in this example is the SKU of the product that was not found in product.txt.

Task Seven

Implement the function `void drawPrices(int numOfProducts, int productSkus[])`.

This function draws a figure that shows how some product prices compare to each other.

It accepts the number of products we wish to compare, and an array of SKUs of the products to compare.

Each product will be represented by a black bar that has width 30. The length of the bar will be equal to ceiling(price of product) x 10. Bars are separated from each other by 20 white pixels.

The picture should have height equal to 500 pixels, and width equal to
(30 x numOfProducts + 20 x numOfProducts + 1) pixels.

Check figure 5 for more details. It contains a drawing for 8 products (the 8^{th} does not appear in the picture because it does not exist in products.txt so it was ignored).

Note 1: If the price is of a product is on discount, use the discounted price.
Note 2: If any provided SKU is not in the list of products, PRINT an error message as follows: "SKU 58312 is not available! Ignoring!", where 58312 in this example is the SKU of the product that was not found in product.txt.

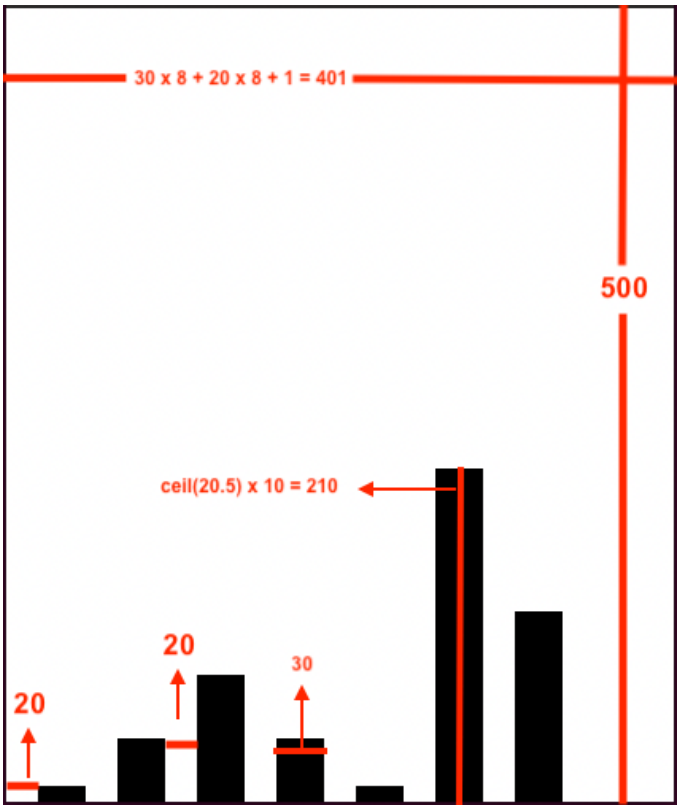

Figure 5

Task Eight

Suppose the store wanted to send information about the product prices and the customer receipts to another computer for archiving purposes. Suppose your computer is at point A and the other computer is at point B, and answer the following questions:

What type of connection do you prefer to use to connect the 2 computers?

Which communication protocol would you use? Identify **all** the available communication protocols that can be used to implement the connection, then select one of them and justify your selection. Use a block diagram to support your answer.

THE END

جامعة الحسين التقنية
University

STUDENT ASSESSMENT SUBMISSION AND DECLARATION

When submitting evidence for assessment, each student must sign a declaration confirming that the work is their own.

Student name: **Student ID:**		**Assessor name:**
Issue date:	**Submission date:** / /2022	**Submitted on:** / /2022
Programme:		
HTU Course Name: Fundamentals of Computing **BTEC UNIT Title *: NA** **HTU Course Code:** 30201100 **BTEC UNIT Code: NA** **I AM REPEATING THIS UNIT*: (YES) (NO)**		

Plagiarism

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalised. It is your responsibility to ensure that you understand **correct referencing practices**. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student signature: **Date:**

جامعة الحسين التقنية
Al Hussein Technical University

Learning Outcomes and Assessment Criteria			
Learning Outcome	**Pass**	**Merit**	**Distinction**
LO1: Discuss the basic concepts of computer hardware, software and operating systems, with the basic relation between them, taking C commands as an example.	**P1** Identify the basic hardware of the computer and how to deal with it. **P2** Identify the basic software of the computer, and able to interact with Ubuntu operating system through its terminal.	**M1** Analyse the basic structure of simple C program. **M2** Express any arithmetic or logical expression to its equivalent in C language. **M3** Apply C language programming basic concepts to trace and debug simple C programs.	**D1** Critically analyse basic C commands and relate it with the machine cycle.
LO2: Implement a full program in C with identification of the three kinds of control structures: sequence, selection, and repetition, understanding the capabilities of implementing C codes that capable to deal with functions.	**P3** Implement problem solving decisions in algorithms and introduce structure charts as a system documentation tool. **P4** Identify the C syntax of sequence, selection, and repetition and when to use each statement type. **P5** Identify the structure of functions with passing information and get data if needed.	**M4** Solve real life problems using control structures in C language integrated with functions.	**LO 2 & 3** **D2** Critically evaluate the source code that solve real life problem implementing the functions, arrays, and files following the code standards and best practices.
LO3: Understanding the basic concepts of pointers in C with implementation of arrays and dealing with files. In addition to the role of C programming as a powerful tool to communicate between devices and manipulate their data.	**P6** Explore pointer's role in referencing variables, and how it is reflected in the memory. **P7** Implement arrays as one dimensional or two dimensional and dealing with any operation on them. **P8** Explore the fundamentals of data communication between devices.	**M5** Implement the effectiveness of arrays in real applications with usage of functions. **M6** Identify streams in C and their relationship to deal with files. **M7** Introduce a block diagram to solve a real-life problem in communication between devices as a practical application of C language.	