

## 1. Introduction

Congratulations! You are finally launching the business you've been planning for years – an online PC shop! This shop will allow customers to buy from one of two main categories: *custom builds* (a computer that is built and customized to meet specific customer needs) and *pre-built machines* (complete systems assembled by well-known manufacturers using balanced hardware configurations). You currently only carry a relatively small (but powerful) inventory of items, but you want to give your customers the best experience possible anyways. You will be writing a program (using Python) that allows users to order items from your shop.

In this assignment, you will get practice with:

- Creating functions
- Calling functions and using returned values
- User input in a loop
- Conditionals
- Lists, 2-D lists, and tuples
- String formatting for clean output

To begin, let's note a few things. If your customers are building a computer, they absolutely **must** have these essential parts:

1. Central Processing Unit (**CPU**) for executing instructions
2. **Motherboard** for allowing communication between crucial components of the system
3. Random Access Memory (**RAM**) for storing information that needs to be retrieved quickly
4. Power Supply Unit (**PSU**) for powering the components within the system
5. Either a Solid State Drive (**SSD**) or a Hard Disk Drive (**HDD**) (or both) for storage
6. A **case** for storing the components

Often, people will also want a **Graphics Card** to add to their build if they plan on doing things like gaming, video editing, or 3D graphics modeling in which processing graphical data as efficiently as possible is important.

With that out of the way, let's also note that sometimes certain PC parts are not compatible with each other within a build (for example, an Intel i7-11700K CPU and an MSI B550-A PRO motherboard are **not** compatible) and the PC will not work when a build is attempted. Your shop tries to ensure that all components can be used together in a build, however sometimes it's just not possible.

**\*\* Note that you do **not** need to know about component compatibility beyond what is explicitly being told to you for this assignment.**

## 2. Files

For this assignment, you must create one file: **shop.py**. Note that this file should be shop.py precisely (not Shop.py, shop.py.py, or any variation of these things).

## 3. Instructions

In the shop.py file, create a function called **pickItems()** which takes no parameters and acts as the heart of the program. This function will mainly consist of a user-input-driven system for ordering PCs, but can (and definitely should) get help from separate helper functions (you'll decide what those helper functions are and what they do). The pickItems() function will walk the user through their order using print statements and collecting user input, and **will ultimately return a *list with the prices of each PC a user has purchased (whether they built it themselves or bought a pre-built)***.

### 3.1 Storing your inventory as a list of lists

You have a static inventory (we'll assume you have *unlimited* resources of each offered item). The types of PC components you have in stock for custom builds are listed below (and can be copied directly into your shop.py file):

```
SSD = [['1', '250 GB', 69.99], ['2', '500 GB', 93.99], ['3', '4 TB', 219.99]]
HDD = [['1', '500 GB', 106.33], ['2', '1 TB', 134.33]]
CPU = [['1', 'Intel Core i7-11700K', 499.99], ['2', 'AMD Ryzen 7 5800X', 312.99]]
MOTHERBOARD = [['1', 'MSI B550-A PRO', 197.46], ['2', 'MSI Z490-A PRO', 262.30]]
RAM = [['1', '16 GB', 82.99], ['2', '32 GB', 174.99]]
GRAPHICS_CARD = [['1', 'MSI GeForce RTX 3060 12GB', 539.99]]
PSU = [['1', 'Corsair RM750', 164.99]]
CASE = [['1', 'Full Tower (black)', 149.99], ['2', 'Full Tower (red)', 149.99]]
```

And the types of pre-built machines you have to offer are:

```
PREBUILTS = [['1', 'Legion Tower Gen 7 with RTX 3080 Ti', 3699.99], ['2', 'SkyTech Prism II Gaming PC', 2839.99], ['3', 'ASUS ROG Strix G10CE Gaming PC', 1099.99]]
```

In these 2-D lists, the inner-list structure represents:

[the **ID** associated with that component (string), the **name** of the product being offered (string), and the **price** of the product (float)]

### 3.2 Hardware Incompatibilities

As mentioned previously, some components are incompatible with each other. For example, some motherboards do not have the correct CPU socket (and supported chipset) for certain CPU models (this is all jargon you don't need to be familiar with, but could research if you were interested).

When looking at the hardware we offer, we notice that the Intel Core i7-11700K CPU is **NOT** compatible with the MSI B550-A PRO motherboard, and the AMD Ryzen 7 5800X CPU is **NOT** compatible with the MSI Z490-A PRO motherboard. This needs to be taken into consideration when showing customers what components are available for their build.

If the user is building a custom machine and chooses option '1' for their CPU (Intel Core i7-11700K), your program should only present and allow them to select option '2' (MSI Z490-A PRO) to them within the motherboard selection. If the user selects option '2' for their CPU (AMD Ryzen 7 5800X), the program should only present and allow them to select option '1' (MSI B550-A PRO) within the motherboard selection stage. Examples of this are shown below.

```
First, let's pick a CPU.  
1 : Intel Core i7-11700K, $499.99  
2 : AMD Ryzen 7 5800X, $312.99  
Choose the number that corresponds with the part you want: 1  
  
Next, let's pick a compatible motherboard.  
2 : MSI Z490-A PRO, $262.30  
Choose the number that corresponds with the part you want: 
```

```
First, let's pick a CPU.  
1 : Intel Core i7-11700K, $499.99  
2 : AMD Ryzen 7 5800X, $312.99  
Choose the number that corresponds with the part you want: 2  
  
Next, let's pick a compatible motherboard.  
1 : MSI B550-A PRO, $197.46  
Choose the number that corresponds with the part you want: 
```

### 3.3 Storage Selection

When purchasing storage (where data is permanently stored on your computer), the customer can either select an HDD, an SSD, or **both**. Thus, you must allow them to decline purchasing the HDD or the SSD if they wish. To 'decline', the user should be able to input 'x' OR 'X'. If the user has already declined purchasing an SSD, you must ensure that they select a HDD. Examples of this are shown below.

```
Next, let's pick an SSD (optional, but you must have at least one SSD or HDD).
1 : 250 GB, $69.99
2 : 500 GB, $93.99
3 : 4 TB, $219.99
Choose the number that corresponds with the part you want (or X to not get an SSD): x

Next, let's pick an HDD (optional, but you must have at least one SSD or HDD).
1 : 500 GB, $106.33
2 : 1 TB, $134.33
Choose the number that corresponds with the part you want (since you did not get an SSD, you must get an HDD): x
Choose the number that corresponds with the part you want (since you did not get an SSD, you must get an HDD): 1
```

```
Next, let's pick an SSD (optional, but you must have at least one SSD or HDD).
1 : 250 GB, $69.99
2 : 500 GB, $93.99
3 : 4 TB, $219.99
Choose the number that corresponds with the part you want (or X to not get an SSD): 3

Next, let's pick an HDD (optional, but you must have at least one SSD or HDD).
1 : 500 GB, $106.33
2 : 1 TB, $134.33
Choose the number that corresponds with the part you want (or X to not get an HDD): X
```

### 3.4 Order of Component Selection

It is important to note that customers **MUST** purchase one CPU, one motherboard, one RAM option, one PSU option, one SSD **or** HDD (or **BOTH** – they can decline purchasing one or the other if they wish), and one case. In addition, recall that they do not **NEED** to purchase a graphics card – they can decline with 'x' or 'X'.

Once your program is launched, If the user wishes to build a custom PC, they must enter 1. Then, you should guide them through their options in the exact order of: CPU, Motherboard (making sure the motherboards shown are compatible with the CPU they have just chosen), **RAM, PSU, the case, SSD, HDD, and Graphics card**. This ensures that, during testing, our input is matching up with what your program is expecting. If you do not prompt the user to select the components in this order, **the autograder will fail all of the test cases and you will receive a mark of 0 on that portion.**

### 3.5 Ensuring the Customer Enters a Valid Input

When the user selects an invalid option (one that does not appear as available for selection), the program should continue prompting for input until their entry is valid, as shown in the image below. We can see here that the program continues asking for input until a valid option is chosen.

```
First, let's pick a CPU.  
1 : Intel Core i7-11700K, $499.99  
2 : AMD Ryzen 7 5800X, $312.99  
Choose the number that corresponds with the part you want: 2  
Valid input  
Next, let's pick a compatible motherboard.  
1 : MSI B550-A PRO, $197.46  
Choose the number that corresponds with the part you want: 2  
Choose the number that corresponds with the part you want: 2  
Choose the number that corresponds with the part you want: 3  
Choose the number that corresponds with the part you want: X  
Choose the number that corresponds with the part you want: x  
Choose the number that corresponds with the part you want: 1  
  
Next, let's pick your RAM.
```

When choosing a graphics card, ensure that 'x' and 'X' are included in the allowable input (as the user does not necessarily need one).

When choosing storage, ensure that 'x' and 'X' are included in allowable input for the HDD if the user selected an SSD. If the user did not select an SSD, ensure that 'x' and 'X' are not included in the allowable input.

For all other necessary hardware, the user should **not be able** to select 'x' or 'X' (as these parts are required).

### 3.6 What is Returned from pickItems()?

When building a custom-build, once all of the parts are successfully picked, the total price of all the parts combined should be added to a list that will eventually be returned from pickItems() (once the user selects that they're ready to checkout with '3'). The user, if they wish, can also continue to either build another PC (by pressing '1'), or add a pre-built PC to their cart (by pressing '2').

The pickItems() function should return a list of prices of each PC ordered by one customer **in the order that they were purchased**. Again, one customer could order as many custom-built and pre-built machines as they wish (before pressing '3' to checkout) and the total price of each should be returned in a list in the order that the customer purchased them once they do press '3' to checkout.

Note that if the user does not order anything, an empty list should be returned.

### 3.7 The Final Print Statement

After the user selects '3' to checkout, you should simply print the list that is returned from pickItems().

### 3.8 Other Important Notes

After completing a custom-build, you should print 'You have selected all of the required parts! Your total for this PC is \$\_\_\_\_' where '\_\_\_\_' is replaced with the price for that specific PC. If a pre-built is added, you should print 'Your total price for this pre-built is \$\_\_\_\_' where '\_\_\_\_' is replaced with the price for that specific pre-built.

This allows the user to know how much every individual PC costs before checking out.

## 4. User Interaction

The following five images demonstrate the expected flow of the program.

```
Welcome to my PC shop!  
  
Would you like to build a custom PC (1), purchase a pre-built PC (2), or would you like to checkout (3)? 3  
[]
```

```
Welcome to my PC shop!  
  
Would you like to build a custom PC (1), purchase a pre-built PC (2), or would you like to checkout (3)? 2  
  
Great! Let's pick a pre-built PC!  
  
Which prebuilt would you like to order?  
1 : Legion Tower Gen 7 with RTX 3080 Ti, $3699.99  
2 : SkyTech Prism II Gaming PC, $2839.99  
3 : ASUS ROG Strix G10CE Gaming PC, $1099.99  
Choose the number that corresponds with the part you want: 500  
Choose the number that corresponds with the part you want: X  
Choose the number that corresponds with the part you want: 3  
  
Your total price for this prebuilt is $1099.99  
  
Would you like to build a custom PC (1), purchase a pre-built PC (2), or would you like to checkout (3)? 3  
[1099.99]
```



```
Welcome to my PC shop!

Would you like to build a custom PC (1), purchase a pre-built PC (2), or would you like to checkout (3)? 1

Great! Let's start building your PC!

First, let's pick a CPU.
1 : Intel Core i7-11700K, $499.99
2 : AMD Ryzen 7 5800X, $312.99
Choose the number that corresponds with the part you want: 1

Next, let's pick a compatible motherboard.
2 : MSI Z490-A PRO, $262.30
Choose the number that corresponds with the part you want: 2

Next, let's pick your RAM.
1 : 16 GB, $82.99
2 : 32 GB, $174.99
Choose the number that corresponds with the part you want: 2

Next, let's pick your PSU.
1 : Corsair RM750, $164.99
Choose the number that corresponds with the part you want: 1

Next, let's pick your case.
1 : Full Tower (black), $149.99
2 : Full Tower (red), $149.99
Choose the number that corresponds with the part you want: 2

Next, let's pick an SSD (optional, but you must have at least one SSD or HDD).
1 : 250 GB, $69.99
2 : 500 GB, $93.99
3 : 4 TB, $219.99
Choose the number that corresponds with the part you want (or X to not get an SSD): 3

Next, let's pick an HDD (optional, but you must have at least one SSD or HDD).
1 : 500 GB, $106.33
2 : 1 TB, $134.33
Choose the number that corresponds with the part you want (or X to not get an HDD): 1

Finally, let's pick your graphics card (or X to not get a graphics card).
1 : MSI GeForce RTX 3060 12GB, $539.99
Choose the number that corresponds with the part you want: X

You have selected all of the required parts! Your total for this PC is $1578.58

Would you like to build a custom PC (1), purchase a pre-built PC (2), or would you like to checkout (3)? 3
[1578.58]
```

```
Welcome to my PC shop!

Would you like to build a custom PC (1), purchase a pre-built PC (2), or would you like to checkout (3)? 5

Would you like to build a custom PC (1), purchase a pre-built PC (2), or would you like to checkout (3)? 1

Great! Let's start building your PC!

First, let's pick a CPU.
1 : Intel Core i7-11700K, $499.99
2 : AMD Ryzen 7 5800X, $312.99
Choose the number that corresponds with the part you want: 3
Choose the number that corresponds with the part you want: 2

Next, let's pick a compatible motherboard.
1 : MSI B550-A PRO, $197.46
Choose the number that corresponds with the part you want: 1

Next, let's pick your RAM.
1 : 16 GB, $82.99
2 : 32 GB, $174.99
Choose the number that corresponds with the part you want: 1

Next, let's pick your PSU.
1 : Corsair RM750, $164.99
Choose the number that corresponds with the part you want: 1

Next, let's pick your case.
1 : Full Tower (black), $149.99
2 : Full Tower (red), $149.99
Choose the number that corresponds with the part you want: 2

Next, let's pick an SSD (optional, but you must have at least one SSD or HDD).
1 : 250 GB, $69.99
2 : 500 GB, $93.99
3 : 4 TB, $219.99
Choose the number that corresponds with the part you want (or X to not get an SSD): 5
Choose the number that corresponds with the part you want (or X to not get an SSD): 3

Next, let's pick an HDD (optional, but you must have at least one SSD or HDD).
1 : 500 GB, $106.33
2 : 1 TB, $134.33
Choose the number that corresponds with the part you want (or X to not get an HDD): 1

Finally, let's pick your graphics card (or X to not get a graphics card).
1 : MSI GeForce RTX 3060 12GB, $539.99
Choose the number that corresponds with the part you want: 1

You have selected all of the required parts! Your total for this PC is $1774.73

Would you like to build a custom PC (1), purchase a pre-built PC (2), or would you like to checkout (3)? 2

Great! Let's pick a pre-built PC!

Which prebuilt would you like to order?
1 : Legion Tower Gen 7 with RTX 3080 Ti, $3699.99
2 : SkyTech Prism II Gaming PC, $2839.99
3 : ASUS ROG Strix G10CE Gaming PC, $1099.99
Choose the number that corresponds with the part you want: 1

Your total price for this prebuilt is $3699.99

Would you like to build a custom PC (1), purchase a pre-built PC (2), or would you like to checkout (3)? 3
[1774.73, 3699.99]
```

```
Welcome to my PC shop!

Would you like to build a custom PC (1), purchase a pre-built PC (2), or would you like to checkout (3)? 1

Great! Let's start building your PC!

First, let's pick a CPU.
1 : Intel Core i7-11700K, $499.99
2 : AMD Ryzen 7 5800X, $312.99
Choose the number that corresponds with the part you want: 1

Next, let's pick a compatible motherboard.
2 : MSI Z490-A PRO, $262.3
Choose the number that corresponds with the part you want: 2

Next, let's pick your RAM.
1 : 16 GB, $82.99
2 : 32 GB, $174.99
Choose the number that corresponds with the part you want: 1

Next, let's pick your PSU.
1 : Corsair RM750, $164.99
Choose the number that corresponds with the part you want: 1

Next, let's pick your case.
1 : Full Tower (black), $149.99
2 : Full Tower (red), $149.99
Choose the number that corresponds with the part you want: 2

Next, let's pick an SSD (optional, but you must have at least one SSD or HDD).
1 : 250 GB, $69.99
2 : 500 GB, $93.99
3 : 4 TB, $219.99
Choose the number that corresponds with the part you want (or X to not get an SSD): X

Next, let's pick an HDD (optional, but you must have at least one SSD or HDD).
1 : 500 GB, $106.33
2 : 1 TB, $134.33
Choose the number that corresponds with the part you want (since you did not get an SSD, you must get an HDD): 1

Finally, let's pick your graphics card (or X to not get a graphics card).
1 : MSI GeForce RTX 3060 12GB, $539.99
Choose the number that corresponds with the part you want: 1

You have selected all of the required parts! Your total for this PC is $1806.58

Would you like to build a custom PC (1), purchase a pre-built PC (2), or would you like to checkout (3)? 2

Great! Let's pick a pre-built PC!

Which prebuilt would you like to order?
1 : Legion Tower Gen 7 with RTX 3080 Ti, $3699.99
2 : SkyTech Prism II Gaming PC, $2839.99
3 : ASUS ROG Strix G10CE Gaming PC, $1099.99
Choose the number that corresponds with the part you want: 3

Your total price for this prebuilt is $1099.99

Would you like to build a custom PC (1), purchase a pre-built PC (2), or would you like to checkout (3)? 3
[1806.58, 1099.99]
```

**Important Notes:**

- The number that the user inputs needs to be the number in the first index of the item they are selecting, otherwise known as the **ID** (for example, if they are purchasing 32GB of RAM, they need to enter '2'. This is so that when we test your code in Gradescope, our input will match up with the values that your program is looking for.
- If the user can decline selecting an item (for example, the SSD, HDD, or Graphics card), allow them to select 'X' (they should be able to enter a capital 'X' or lowercase 'x') in order to not purchase that piece of hardware.
- The exact text that you print to the user should be as similar as possible to the examples provided above. This makes grading easier.

**Hints:**

- You must ensure that the user is only selecting one of the offered values (for example, for the SSD, those values would be '1', '2', '3', 'x', or 'X'). To do this, you could write a helper function like 'extract(list)' that returns a list of the FIRST element of each list within the list of hardware. So, for SSD, it would return ['1', '2', '3'] (and then you could add 'x' and 'X' manually)

The usage could look something like:

```
while (SSD_pick not in extract(SSD) and SSD_pick != 'x' and SSD_pick != 'X'):
    SSD_pick = input('Choose the number that corresponds with the
part you want (or X to not get an SSD): ')
```

This will keep querying the user for a value until they enter an 'permitted' value.

- When a user is purchasing a HDD, you should check if they have already purchased an SSD (and had not pressed 'x' or 'X'). This can be done with a simple 'if-else' statement. If they have, you can allow 'x' or 'X' to be an allowable input. If they haven't, you should ensure that they choose one of the available HDD options so that the PC will work upon building it.

## Tips and Guidelines

- Add comments throughout your code to explain what each section of the code is doing and/or how it works. This is crucial in obtaining part-marks.
- Ensure that both 'x' and 'X' work as inputs to the program, where they are applicable.
- You are encouraged to create and run your own tests to ensure that your code works in many different cases (you can also use the images provided within this document to determine whether your program is working as expected or not).
- When submitting your assignment on Gradescope, please submit `shop.py` **only**.
- Ensure that when printing prices, you are using **two digits** after the decimal point.

## Rules

- Read and follow the instructions carefully.
- Only submit the python file described in the files section of this document.
- Submit the assignment on time. Late submissions will receive a late penalty of 10% per day (except where late coupons are used).
- Forgetting to submit a finished assignment is **not** a valid excuse for submitting late.
- Submissions must be done on Gradescope. They will not be accepted by email.
- You may re-submit your code as many times as you would like. However, re-submissions that come in after the due date will be considered late and subjected to late penalties.
- Assignments will be run through a similarity checking software to check for code that looks very similar to that of other students. Sharing or copying code in any way is considered plagiarism and may result in a mark of zero (0) on the assignment and/or reported to the Dean's Office. Work is to be done individually.

## Submission

Due: Wednesday, March 8<sup>th</sup>, 2023 at 11:55pm.

Assignments cannot be submitted by email, only through Gradescope.

## Marking Guidelines

The assignment will be marked as a combination of your auto-graded test results, manual grading of your code logic, comments, formatting, style, etc. Below is a breakdown of the marks for this assignment:

[50 marks] Auto-graded Tests

[20 marks] Code logic and completeness

[10 marks] Comments

[10 marks] Code formatting

[10 marks] Meaningful and properly formatted variables

**Total: 100 marks**