



American International University-Bangladesh (AIUB)

Bachelor of Science in Computer Science & Engineering

Spring 22\_23

Section: A

Course: Python

Mid\_Project

**PROJECT TITLE :** Develop a computer lab management application using Python.

**Name:** Nabajit Dey

**ID :** 19-40618-1

**Submitted to :** Akinul Islam Jony

**Submitted on :** 26/02/2023

## **Project Overview:**

The goal of this project is to use Python to create a console-based application for managing computer labs. The program is made to keep track of all the Computers in a lab. The program's functions will include adding, updating, removing, showing all PCs, showing specific PCs, searching for a certain PC, and saving all the PC data in a text file.

## **Project Solution Design:**

Python is intended to be used to implement the project. A console-based application will be used for the program. We will utilize Python's built-in modules like os, sys, and time to create the application.

The application will have a menu that will display all the available options to the user. The user will select an option from the menu to perform a specific task. The menu will have the following options:

**Add a new PC**

**Update an existing PC**

**Remove an existing PC**

**Display all PCs**

**Display individual PC**

**Search for a PC**

**Store PC information in a text file**

**Quit the application**

The application will have a data structure to store the PC information. We will use a list to store the PC information. Each PC will be represented as a dictionary with the following keys:

**PC Number**

**Operating System**

**Status**

The PC Number will be unique for each PC.

The application will have functions to perform each of the tasks mentioned in the menu. We will also handle exceptions in the application to make it more robust.

## Implementation:

### main.py:

class PC:

```
def __init__(self, pc_number, os, status):  
    self.pc_number = pc_number  
    self.os = os  
    self.status = status
```

class Lab:

```
def __init__(self):  
    self.pcs = []
```

def add\_pc(self):

```
    print("Please Provide some Information to Add PC")  
    pc_number = input("Enter PC number: ")  
    if self.check_pc_number(pc_number):  
        return  
    os = input("Enter operating system: ")  
    status = input("Enter status: ")  
    pc = PC(pc_number, os, status)  
    self.pcs.append(pc)  
    print("PC added successfully.")
```

def update\_pc(self):

```
    print("Please Provide some Information to Update Existing PC")
```

```
pc_number = input("Enter PC number to update: ")
```

```
for pc in self.pcs:
```

```
    if pc.pc_number == pc_number:
```

```
        os = input("Enter new operating system: ")
```

```
        status = input("Enter new status: ")
```

```
        pc.os = os
```

```
        pc.status = status
```

```
        print("PC updated successfully.")
```

```
    return
```

```
print("PC not found.")
```

```
def remove_pc(self):
```

```
    print("Please Provide some Information to Remove Existing PC")
```

```
    pc_number = input("Enter PC number to remove: ")
```

```
    for pc in self.pcs:
```

```
        if pc.pc_number == pc_number:
```

```
            self.pcs.remove(pc)
```

```
            print("PC removed successfully.")
```

```
        return
```

```
print("PC not found.")
```

```
def display_all_pcs(self):
```

```
    print("All the PC's Information")
```

```
    for pc in self.pcs:
```

```
        print("PC number: " + pc.pc_number)
```

```
        print("Operating system: " + pc.os)
```

```
        print("Status: " + pc.status)
```

```
def display_pc(self):
```

```
    print("Single PC Information")
```

```
    pc_number = input("Enter PC number to display: ")
```

```
for pc in self.pcs:
    if pc.pc_number == pc_number:
        print("PC number: " + pc.pc_number)
        print("Operating system: " + pc.os)
        print("Status: " + pc.status)
        return
print("PC not found.")
```

```
def search_pc(self):
    print("Search for PC")
    pc_number = input("Enter PC number to search: ")
    for pc in self.pcs:
        if pc.pc_number == pc_number:
            print("PC number: " + pc.pc_number)
            print("Operating system: " + pc.os)
            print("Status: " + pc.status)
            return
    print("PC not found.")
    choice = input("Do you want to add this PC to the lab? (y/n) ")
    if choice == "y":
        self.add_pc()
```

```
def check_pc_number(self, pc_number):
    for pc in self.pcs:
        if pc.pc_number == pc_number:
            choice = input("PC number already exists. Do you want to modify its information? (y/n) ")
            if choice == "y":
                self.update_pc()
            return True
    choice = input("Do you want to remove this PC from the lab? (y/n) ")
    if choice == "y":
```

```
        self.remove_pc()

    return True

return False
```

```
def store_pcs(self):
    print("Store PC's Information")
    filename = input("Enter filename to store PC information: ")
    with open(filename, "w") as file:
        for pc in self.pcs:
            file.write(pc.pc_number + "," + pc.os + "," + pc.status + "\n")
    print("PC information stored successfully.")
```

```
def stop_program(self):
    print("Stopping program")
    exit()
```

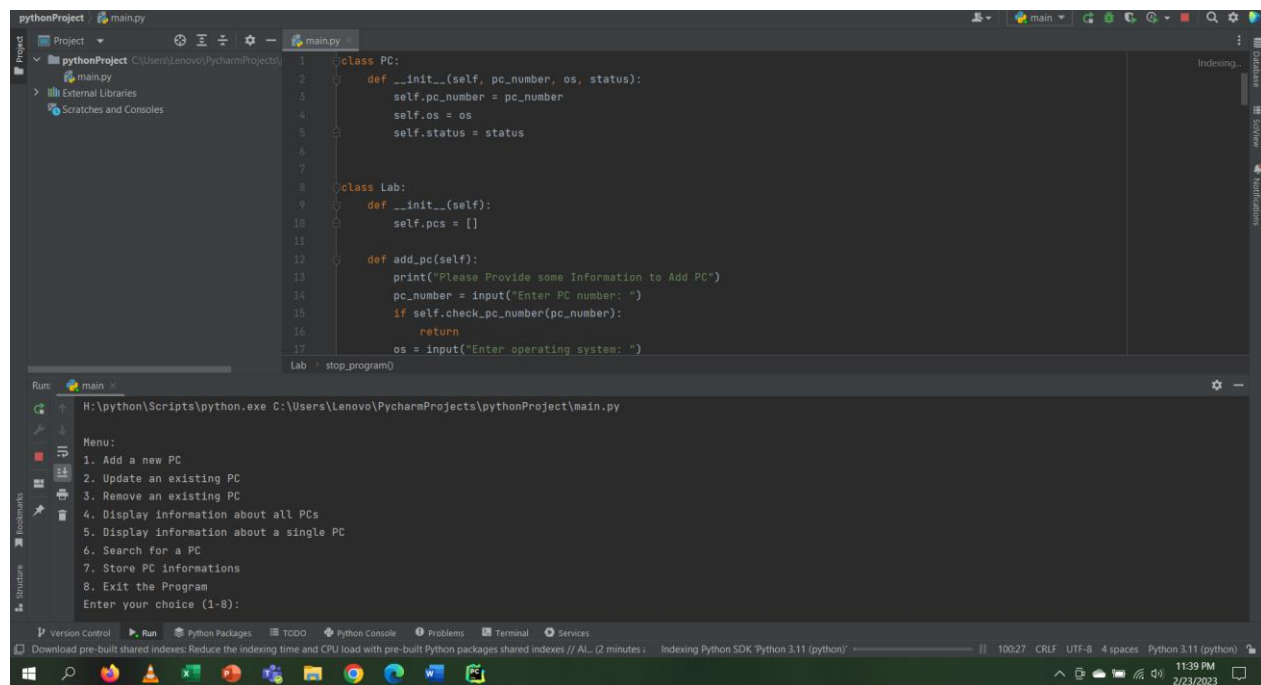
```
lab = Lab()
```

```
while True:
    print("\nMenu:")
    print("1. Add a new PC")
    print("2. Update an existing PC")
    print("3. Remove an existing PC")
    print("4. Display information about all PCs")
    print("5. Display information about a single PC")
    print("6. Search for a PC")
    print("7. Store PC informations")
    print("8. Stop the Program")
    choice = input("Enter your choice (1-8): ")
```

```
if choice == "1":
    lab.add_pc()
elif choice == "2":
    lab.update_pc()
elif choice == "3":
    lab.remove_pc()
elif choice == "4":
    lab.display_all_pcs()
elif choice == "5":
    lab.display_pc()
elif choice == "6":
    lab.search_pc()
elif choice == "7":
    lab.store_pcs()
elif choice == "8":
    lab.stop_program()
else:
    print("Invalid choice. Please try again.")
```

## Project Outcomes:

After run the project



```
pythonProject main.py
1 class PC:
2     def __init__(self, pc_number, os, status):
3         self.pc_number = pc_number
4         self.os = os
5         self.status = status
6
7
8 class Lab:
9     def __init__(self):
10        self.pcs = []
11
12    def add_pc(self):
13        print("Please Provide some Information to Add PC")
14        pc_number = input("Enter PC number: ")
15        if self.check_pc_number(pc_number):
16            return
17        os = input("Enter operating system: ")
18
19 Lab - stop_program()
```

Run: main

H:\python\Scripts\python.exe C:\Users\Lenovo\PycharmProjects\pythonProject\main.py

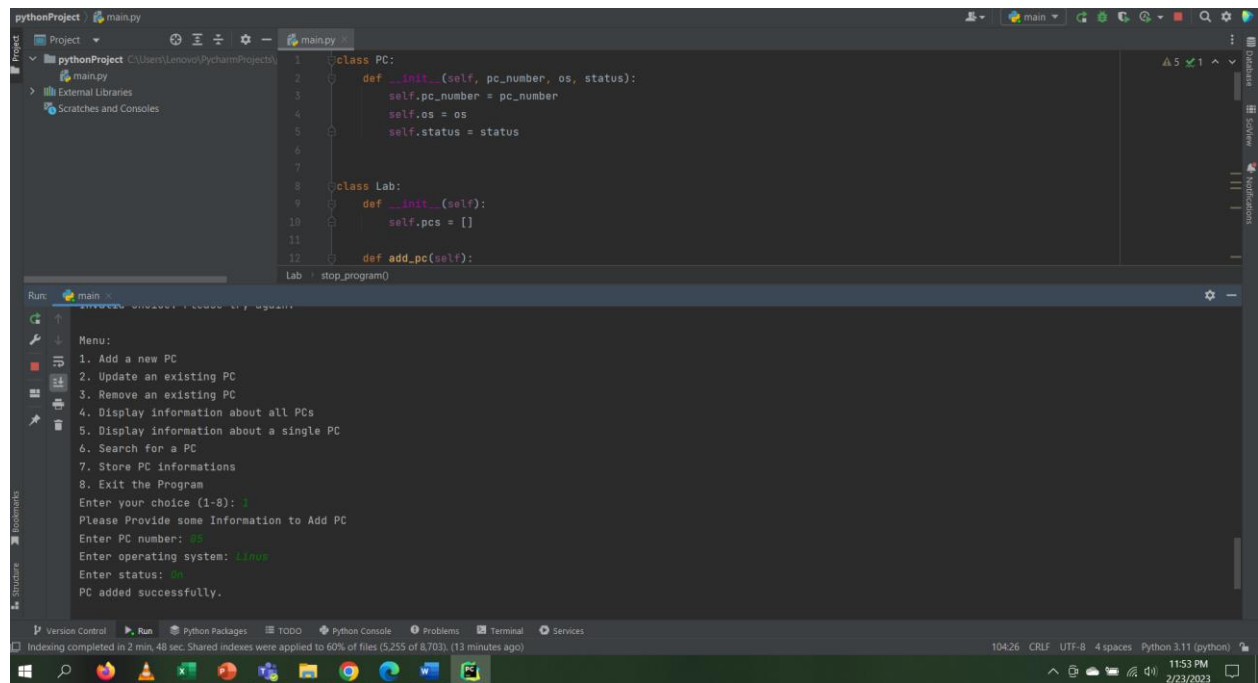
Menu:

1. Add a new PC
2. Update an existing PC
3. Remove an existing PC
4. Display information about all PCs
5. Display information about a single PC
6. Search for a PC
7. Store PC informations
8. Exit the Program

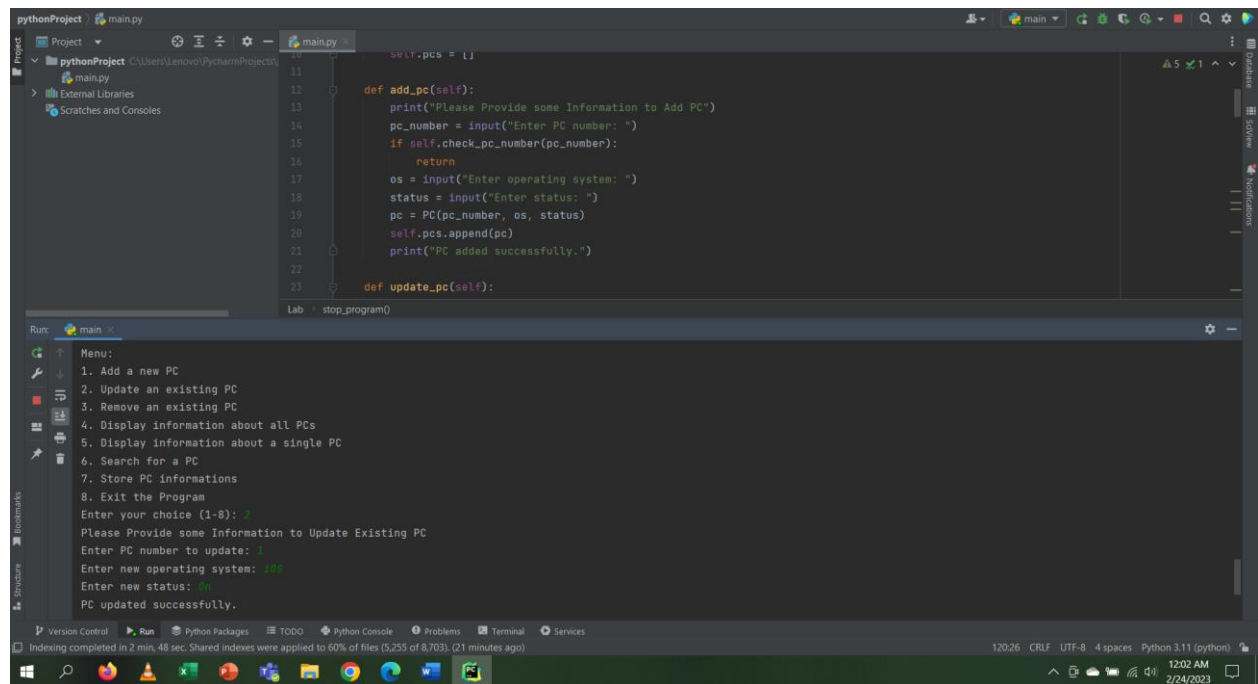
Enter your choice (1-8):

Add a new PC

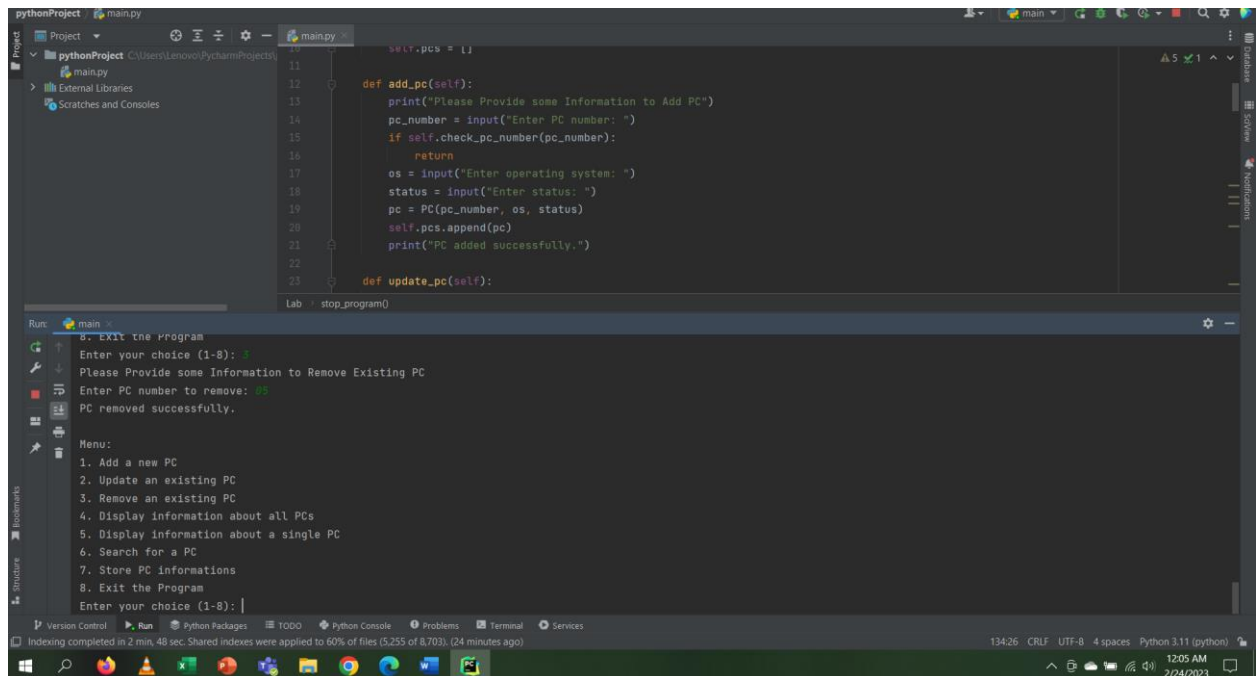




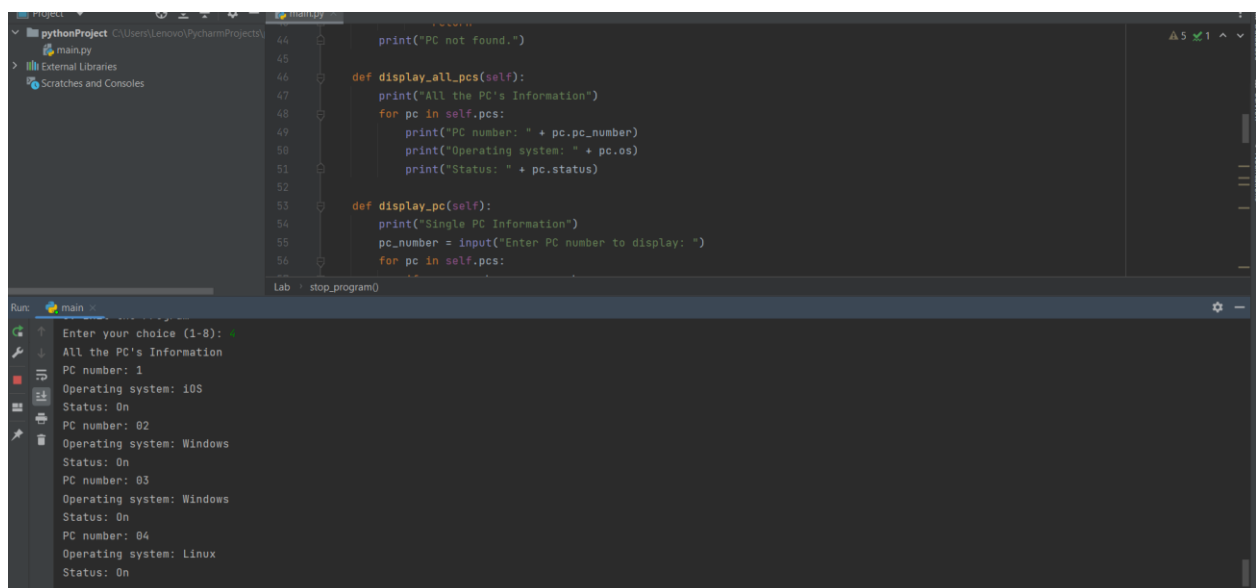
## Update an existing PC



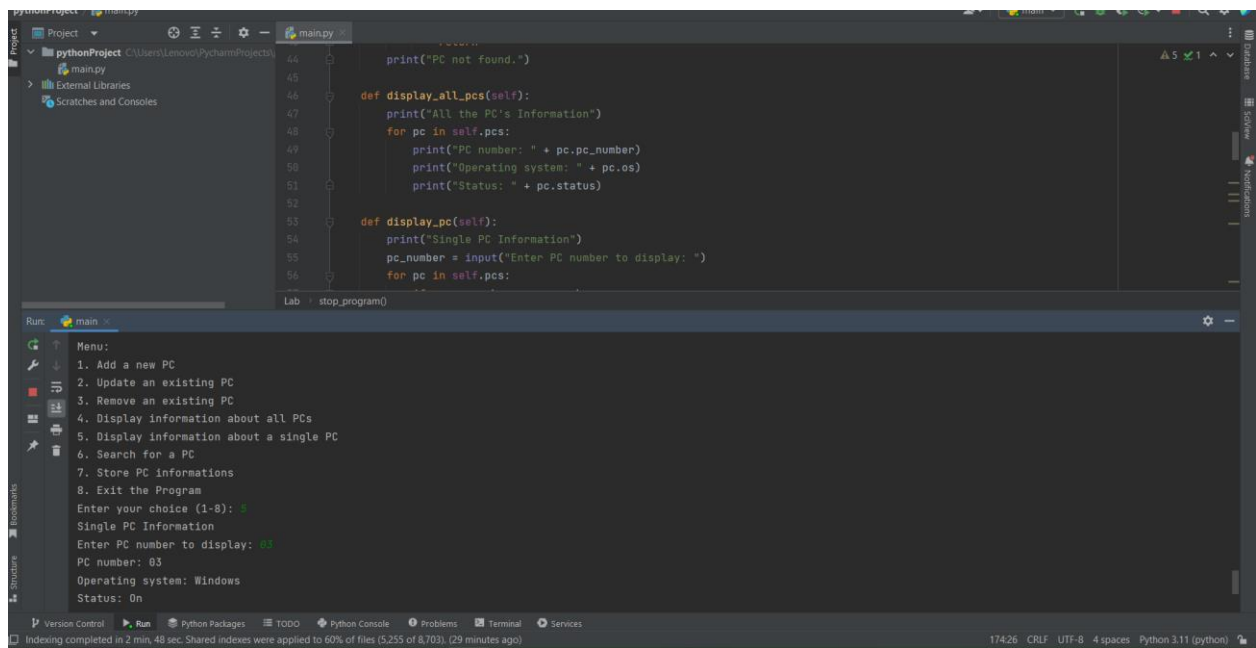
## Remove an existing PC



## Display information about all PCs



## Display information about a single PC

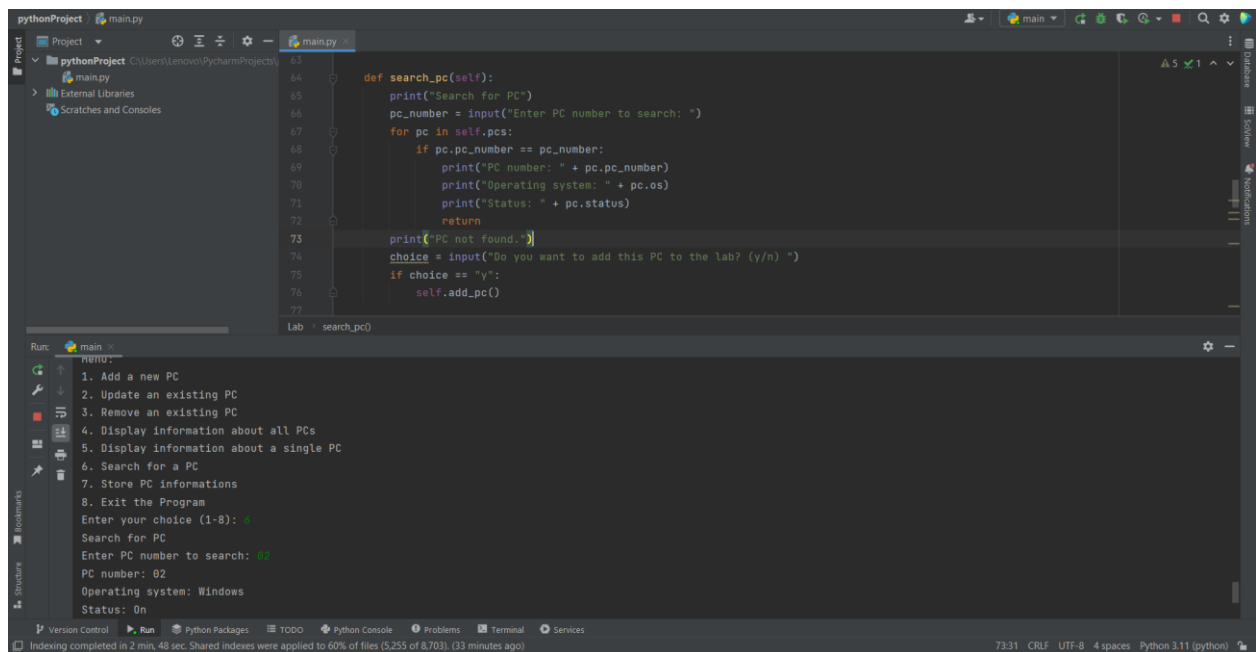


The screenshot shows the PyCharm IDE with the `main.py` file open. The `display_pc` method is highlighted, which prompts the user to enter a PC number to display. The Run console shows the program's output, including a menu, a choice of 5, and the details for PC number 03.

```
44 print("PC not found.")
45
46 def display_all_pcs(self):
47     print("All the PC's Information")
48     for pc in self.pcs:
49         print("PC number: " + pc.pc_number)
50         print("Operating system: " + pc.os)
51         print("Status: " + pc.status)
52
53 def display_pc(self):
54     print("Single PC Information")
55     pc_number = input("Enter PC number to display: ")
56     for pc in self.pcs:
```

```
Run: main
Menu:
1. Add a new PC
2. Update an existing PC
3. Remove an existing PC
4. Display information about all PCs
5. Display information about a single PC
6. Search for a PC
7. Store PC informations
8. Exit the Program
Enter your choice (1-8): 5
Single PC Information
Enter PC number to display: 03
PC number: 03
Operating system: Windows
Status: On
```

## Search for a PC

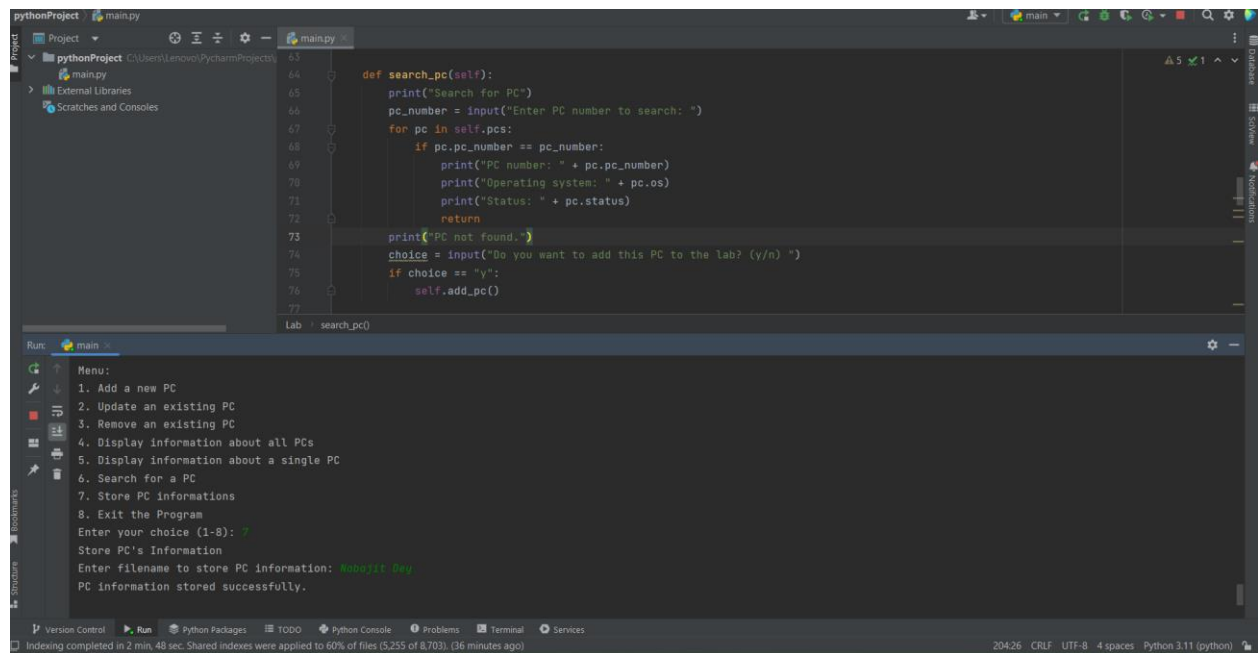


The screenshot shows the PyCharm IDE with the `main.py` file open. The `search_pc` method is highlighted, which prompts the user to enter a PC number to search for. The Run console shows the program's output, including a menu, a choice of 6, and the details for PC number 02.

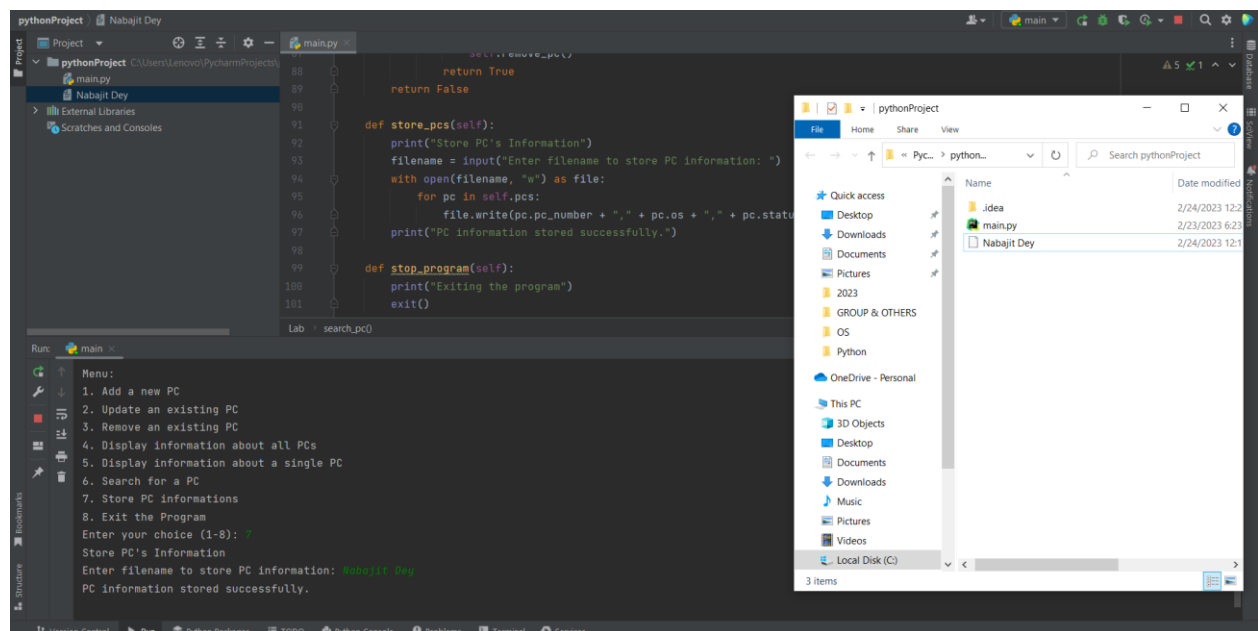
```
63
64
65 def search_pc(self):
66     print("Search for PC")
67     pc_number = input("Enter PC number to search: ")
68     for pc in self.pcs:
69         if pc.pc_number == pc_number:
70             print("PC number: " + pc.pc_number)
71             print("Operating system: " + pc.os)
72             print("Status: " + pc.status)
73             return
74     print("PC not found.")
75     choice = input("Do you want to add this PC to the lab? (y/n) ")
76     if choice == "y":
77         self.add_pc()
```

```
Run: main
Menu:
1. Add a new PC
2. Update an existing PC
3. Remove an existing PC
4. Display information about all PCs
5. Display information about a single PC
6. Search for a PC
7. Store PC informations
8. Exit the Program
Enter your choice (1-8): 6
Search for PC
Enter PC number to search: 02
PC number: 02
Operating system: Windows
Status: On
```

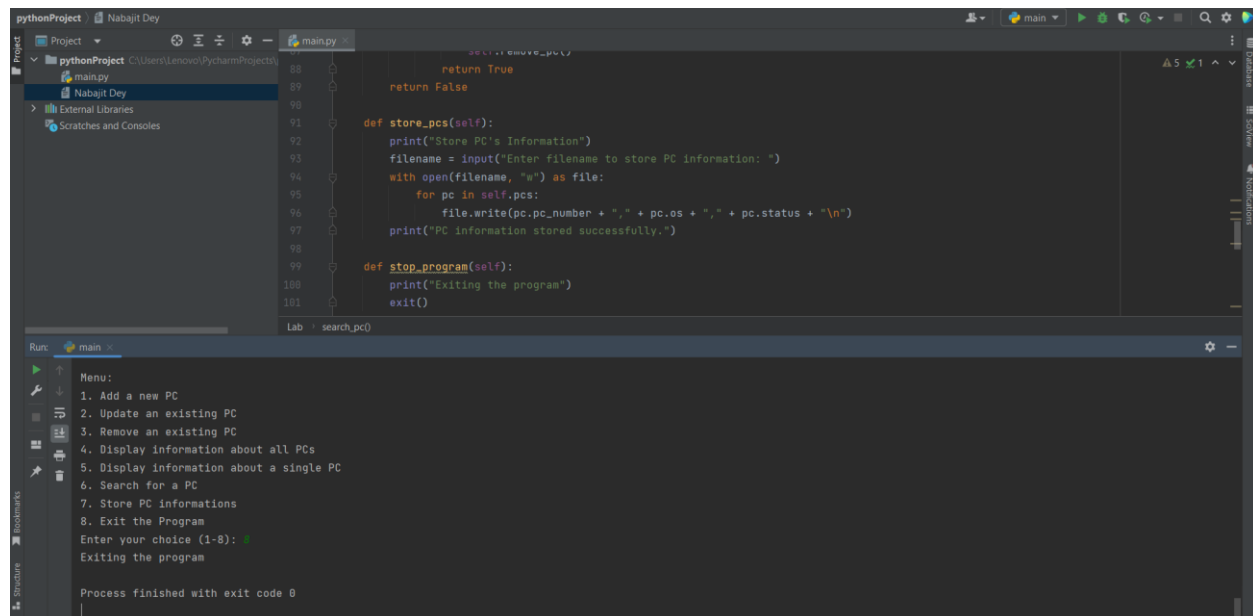
## Store PC informations



## Store PC informations



## Exit the Program



The screenshot shows the PyCharm IDE interface. The main editor displays a Python script named `main.py` with the following code:

```
88         self.remove_pc()
89         return True
90     return False
91
92     def store_pcs(self):
93         print("Store PC's Information")
94         filename = input("Enter filename to store PC information: ")
95         with open(filename, "a") as file:
96             for pc in self.pcs:
97                 file.write(pc.pc_number + "," + pc.os + "," + pc.status + "\n")
98             print("PC information stored successfully.")
99
100     def stop_program(self):
101         print("Exiting the program")
102         exit()
```

The Run console at the bottom shows the output of the program:

```
Run: main
Menu:
1. Add a new PC
2. Update an existing PC
3. Remove an existing PC
4. Display information about all PCs
5. Display information about a single PC
6. Search for a PC
7. Store PC informations
8. Exit the Program
Enter your choice (1-8): 8
Exiting the program

Process finished with exit code 0
```

## Application Overview:

The Computer Lab Management Application developed using Python is a console-based application that allows the user to manage the PCs in a computer lab. The application provides the following functionalities:

**Add a new PC:** The user can add a new PC to the lab by providing the PC number, operating system, and status of the PC.

**Update PC information:** The user can update the information of an existing PC by providing the PC number and the new information.

**Remove an existing PC:** The user can remove an existing PC from the lab by providing the PC number.

**Display all PCs:** The user can view all the PCs in the lab along with their information.

**Display PC information:** The user can view the information of a specific PC by providing the PC number.

**Search for a PC:** The user can search for a specific PC by providing the PC number. If the PC is not found, the user can choose to add a new PC.

**Store PC information to a file:** The user can store all the PC information in a text file.

## **Conclusion:**

In conclusion, the Computer Lab Management Application is a useful tool for managing the PCs in a computer lab. It provides a simple and easy-to-use interface for adding, updating, and removing PCs from the lab, as well as viewing the information of all the PCs or a specific PC. The application can be further improved by adding additional features such as password protection, user authentication, and graphical user interface. Overall, the application is a great example of how Python can be used to develop useful tools for everyday use.

