OOP – Lab #1

Aim: Getting Familiar with Visual Studio and Its Support for Python, Expressions, and Input/Output

Topics:

1. Microsoft Visual Studio IDE (Integrated Development Environment)
2. Python Programming Language
3. Expressions
4. Input/Output in Python

Microsoft Visual Studio - Microsoft Visual Studio is an integrated development environment (IDE) used by developers to write, debug, and test programs.

- It supports multiple programming languages such as C++, C#, Python, and JavaScript.
- It is a powerful tool for creating applications for Windows, web, mobile, and cloud platforms.
- It's an all-in-one platform that simplifies writing, managing, and deploying code.

Installing Microsoft Visual Studio – To install **Microsoft Visual Studio**, follow the steps in Install Visual Studio.

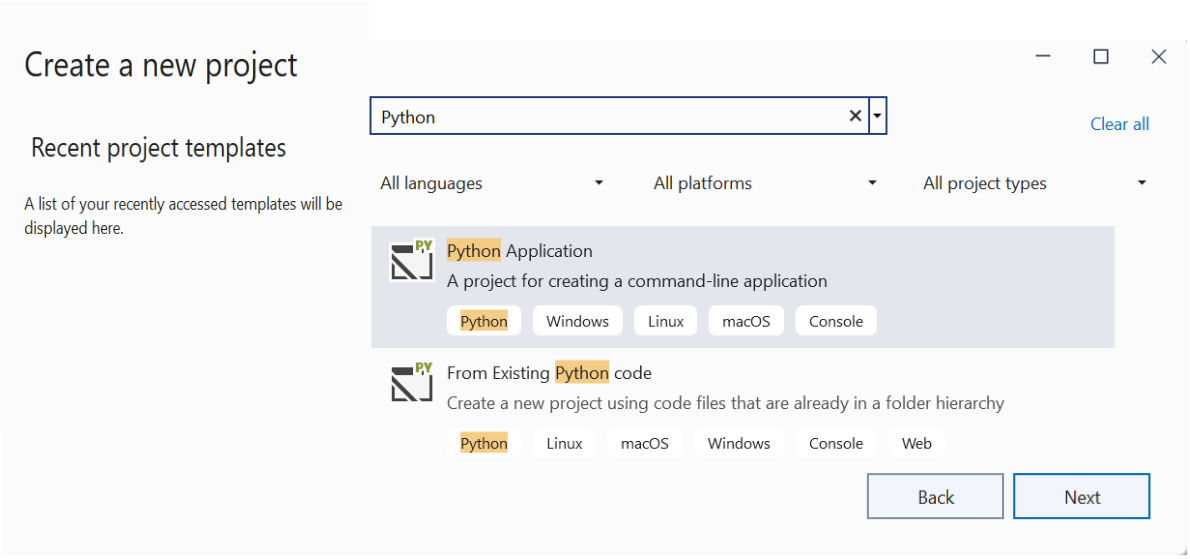
Python Programming Language – Python is a popular programming language that's reliable, flexible, easy to learn, and free to use on all operating systems.

- A strong developer community and many free libraries provide robust support for working with Python.
- The language supports all kinds of development, including web applications, web services, desktop apps, scripting, and scientific computing.
- **Visual Studio** provides first-class language support for Python.

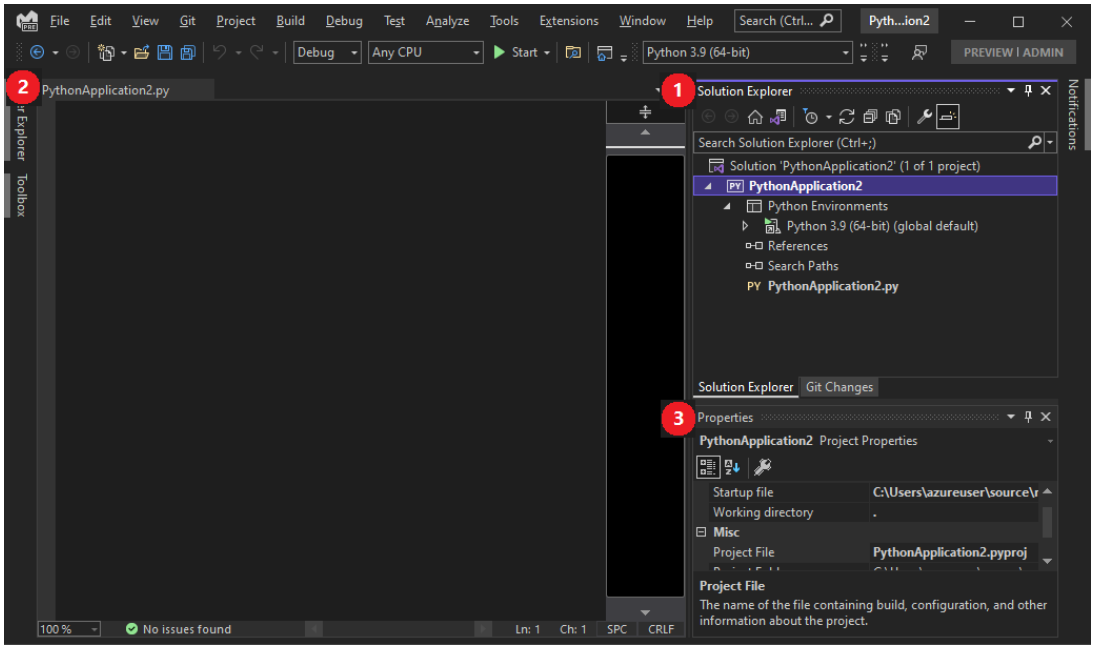
Installing Python Extension in Microsoft Visual Studio – Visual Studio installed with support for Python workloads. For more information, see Install Python Extension in Visual Studio.

Create a New Python Project - A project is how Visual Studio manages all the files that come together to produce a single application. Application files include source code, resources, and configurations.

1. In Visual Studio, select **File > New > Project.**
2. To view Python templates, search for **Python**.
3. Select the **Python Application** template**,** and select **Next.**



4. On the Configure your new project screen, specify a name and file location for the project, and then select **Create**.
5. After a few moments, your new project opens in **Visual Studio:**

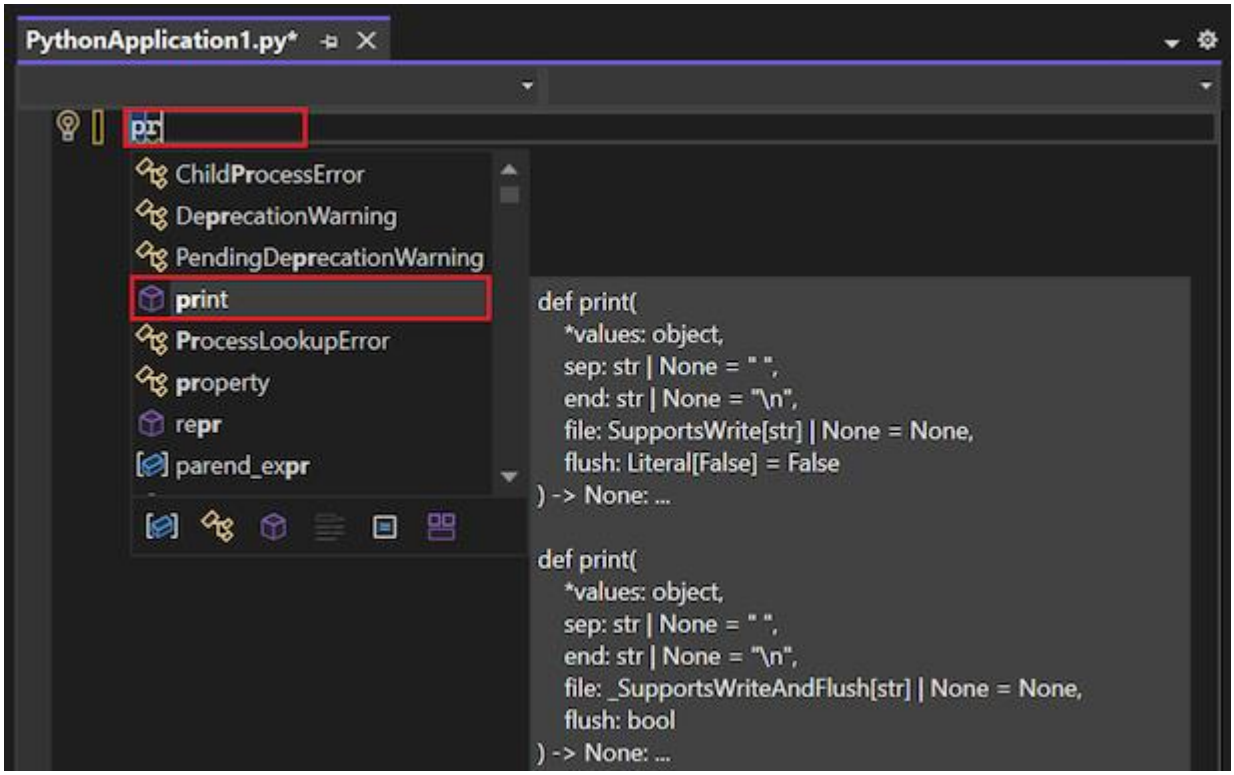


Here's what you see:

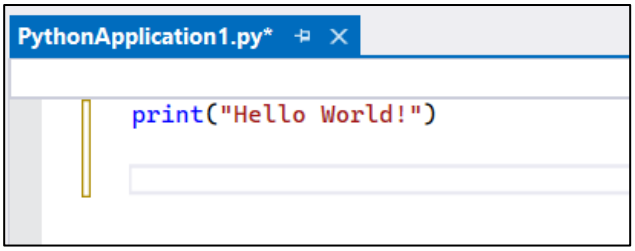
(1): The Visual Studio Solution Explorer window shows the project structure.
(2): The default code file opens in the editor.
(3): The Properties window shows more information for the item selected in Solution Explorer, including its exact location on disk.

Write and Run Python Code in Visual Studio – A Python application project with an empty Python file (.py) is already created.

- When you create a **new Python project** from the **Python Application** template, Visual Studio creates an empty Python file (.py) and opens the file in the **editor**. In this section, you add code to this empty Python file.
- Follow these steps to start writing Python code:
  - Open your empty Python file in the **Visual Studio editor**.
  - In the **editor**, start to enter the Python function name print.



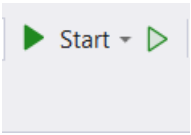
  - In the editor, complete the code statement so it matches the following example:



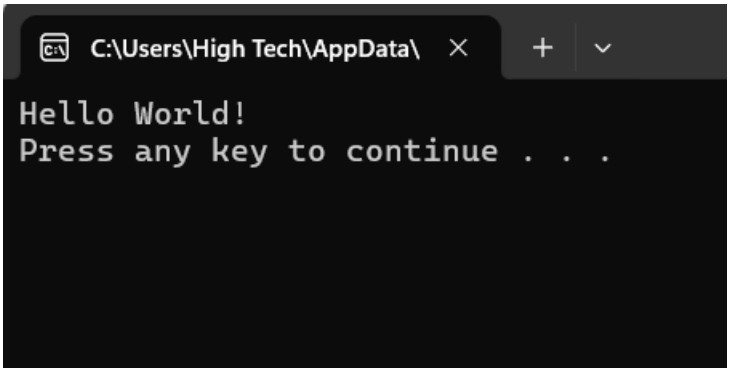
- Visual Studio also uses coloring and formatting techniques to help you read your code.

Run Code in Visual Studio – After you have some code in place, you're ready to try running your program. You can choose to run the application with or without debugging.

- You can run the code **with debugging** or **without debugging**.



- Visual Studio warns you if you still have errors in your code.
- When you run the program, a **console window** opens to show the results.



- To close the console window, press any key. You return to the Visual Studio editor.

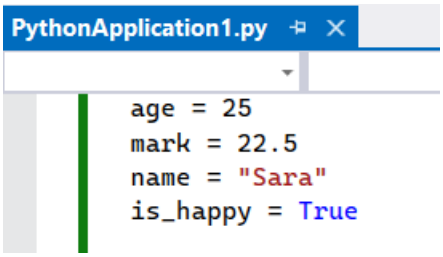
Practices in Python Codes

Basic Object types in Python: Python has different basic object types:

- **int** is used to represent integers (e.g., -3 or 5 or 10002).
- **float** is used to represent real numbers (e.g., 3.0 or 3.17 or -28.72).
- **bool** is used to represent the Boolean values: **True** and **False**.
- **str** is an object in Python Strings.

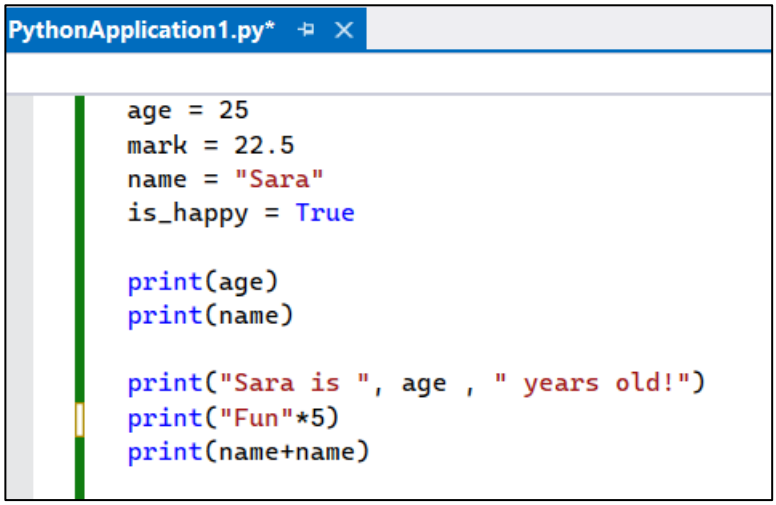
Variables in Python and Assigning Values to Them – In Python, a variable is like **a container that holds a value**, such as a number, a word, or some data you want to use in your program.

- You give the variable a name, and then you can use that name to work with the value stored inside.



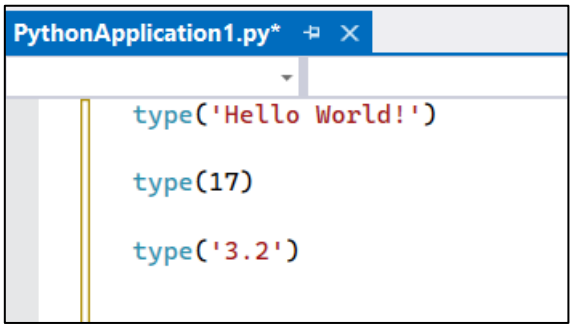
print() Function: The print() function is used to print a specified message or variable value to the screen.

Run all codes to see their output in the Console.



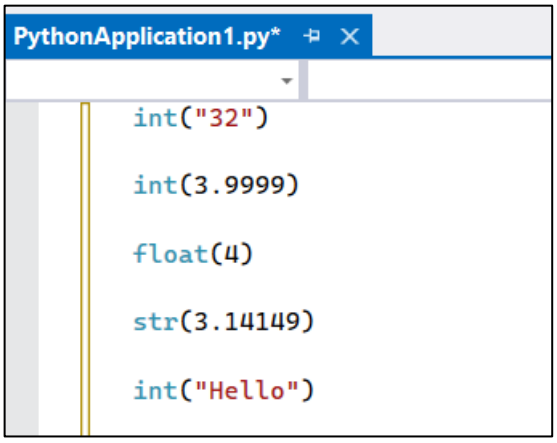
```
a = 3
b = 4

print("the sum is ", a+b)

hour = 2
minutes = 20
print("The total time in minutes: ", hour * 60 + minutes)
```

- **type function:** If you are not sure what type a value has, the interpreter can tell you.



- **Typecasting:** Python defines type conversion functions to directly convert one object type to another:

int (*value***) –** This function converts the object type of *value* into an integer.
float (*value***) –** This function converts the object type of *value* to float.
str (*value***) –** This function converts the object type of *value* to a string.



Question – For which of the above typecasting examples do you face an error? Why?

Lab Questions –

Q1 – Write a Python code to define a variable to store your full name, then print the following:

```
My name is Your Name
from CBS department
in Tishk International University!
```

```
fullName = 'Ali Ameen'

print('My name is', fullName,
      '\nfrom CBS department\nin Tishk International University!')
```

Q2 – Write a Python code to print the sum of 5 and 10 as shown below. Make sure to use variables; two variables to store both numbers and a variable to store their sum result.

```
Sum of  5  and  10  is equal to  15
```

```
a = 5
b = 10

s = a + b

print('Sum of', a, 'and', b, 'is equal to', s)
```

Q3 – Type the following expressions to see their output.

```
print(40/2*3)

print(5**2)

print(2*4**2)

print(8/2)

print(7/2)

print(7//2)

print(14//5)

print(14%10)

print(20%10)
```

Q4 – Write a program to find the result of the following equation, where a = 4 and b = 2.

$$\frac{a^2}{4b}$$

```
a = 4
b = 2

c = (a**2)/(4*b)

print("The result of equation is ", c )
```

Q5 – Write a Python code that asks the user to enter their first name and surname. The program output should be as follows:

```
Please enter your first name: Lana
Please enter your surname: Ahmed
Your full name is:  Lana    Ahmed
```

```
firstName = input("Please enter your first name: ")
surName = input("Please enter your surname: ")

print('Your full name is:', firstName, '\t', surName)
```

Q6 – Write a program to ask the user to enter a rectangle's length and width values. Then calculate the rectangle's area. The program's output is as follows:

```
This program calculates the area of a rectangle!
Enter rectangle's length: 6
Enter rectangle's width: 3
The area of the rectangle is 18
```

```
print('This program calculates the area of a rectangle!')

length = int(input("Enter rectangle's length: "))
width = int(input("Enter rectangle's width: "))
area = length * width

print('The area of the rectangle is', area)
```

Q7 – Write a program that converts a day into hours, minutes, and seconds. Allow the user to input the number of the day(s). Then the program calculates its equivalent hours, minutes, and seconds.

(**Hint:** 1 day = 24 hours, 1 hour = 60 minutes, 1 minute = 60 seconds)

```
Please enter number of days: 2
The 2 day(s) is equal to 48 hours 2880 minutes 172800 seconds
```

```
days = int(input('Please enter number of days: '))

hours = days * 24
minutes = hours * 60
seconds = minutes * 60

print("The", days, "day(s) is equal to", hours, "hours", minutes, "minutes", seconds, "seconds"
```

Q8 – Write a program to let the user enter his/her marks for the quiz, midterm, and final exams. Then calculate the total marks.

(Quizzes out of 20%, mid-term exam out of 30%, final exam out of 50%.)

```
Enter your quiz mark: 10
Enter your midterm mark: 20
Enter your final mark: 40
Your total mark is 70
```

```
quiz = int(input('Enter your quiz mark: '))
midterm = int(input('Enter your midterm mark: '))
final = int(input('Enter your final mark: '))

total = quiz + midterm + final

print('Your total mark is', total )
```