BCA Python Assignment 1 – Student Profile Console App

This PDF contains the completed Mini Project: Student Profile Console App using Python. Student details have been included as requested.

Student Name: Gargi Modi **Program:** BCA Cyber Security **Roll Number:** 2501660035 **Semester:** 1

Completed Python Code:

```
# student_profile.py

# Name: Gargi Modi
# Roll No: 2501660035
# Course: BCA Cyber Security
# Semester: 1st
# Subject: Problem Solving with Python
# Assignment: Unit-1 Mini Project
# Title: Student Profile Console App
# Date:

print("\nWelcome to the Student Profile Console Application!")
print("This tool collects your details and displays a formatted student profile.\n")

full_name = input("Enter your full name: ")
roll_no = input("Enter your Roll Number: ")
program = input("Enter your Program (e.g., BCA): ")
university = input("Enter University Name: ")
city = input("Enter your City: ")
age = int(input("Enter your Age: "))
hobby = input("Enter your Hobby: ")

print("\n--- Operators Demonstration ---")
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))

print("\nArithmetic Operators:")
print("Addition:", num1 + num2)
print("Subtraction:", num1 - num2)
print("Multiplication:", num1 * num2)
print("Division:", num1 / num2)
print("Modulus:", num1 % num2)
print("Exponent:", num1 ** num2)
print("Floor Division:", num1 // num2)

print("\nAssignment Operators:")
x = num1
x += 5
print("num1 += 5:", x)
x -= 3
print("num1 -= 3:", x)

print("\nComparison Operators:")
print("num1 == num2:", num1 == num2)
print("num1 != num2:", num1 != num2)
print("num1 > num2:", num1 > num2)
print("num1 < num2:", num1 < num2)
print("num1 >= num2:", num1 >= num2)
print("num1 <= num2:", num1 <= num2)

print("\nLogical Operators:")
print("Both positive:", num1 > 0 and num2 > 0)
```

```
print("At least one positive:", num1 > 0 or num2 > 0)
print("Not positive num1:", not (num1 > 0))

print("\nIdentity Operators:")
print("num1 is num2:", num1 is num2)
print("num1 is not num2:", num1 is not num2)

sample_str = "Python Programming"
print("\nMembership Operators:")
print("'P' in sample_str:", 'P' in sample_str)
print("'z' not in sample_str:", 'z' not in sample_str)

print("\n--- String Operations ---")
print("Uppercase Name:", full_name.upper())
print("Lowercase City:", city.lower())
print("Title Case University:", university.title())
print("Hobby replaced:", hobby.replace(" ", "-"))
print("Count of 'a' in name:", full_name.lower().count('a'))

print("\nEscape Characters:")
print("Name:\t", full_name)
print("City:\t", city)
print("University:\n", university)

print("\n----------------------------------------")
print("          STUDENT PROFILE SYSTEM        ")
print("----------------------------------------")
print(f"Name:            {full_name}")
print(f"Roll No:         {roll_no}")
print(f"Course:          {program}")
print(f"University:      {university}")
print(f"City:            {city}")
print(f"Age:             {age}")
print(f"Hobby:           {hobby}")
print("----------------------------------------")
print("Welcome to Python Programming!")
print("----------------------------------------\n")

save = input("Do you want to save your profile? (yes/no): ").lower()

if save == "yes":
    with open("student_profile.txt", "w") as f:
        f.write("----- Student Profile -----\n")
        f.write(f"Name: {full_name}\n")
        f.write(f"Roll No: {roll_no}\n")
        f.write(f"Course: {program}\n")
        f.write(f"University: {university}\n")
        f.write(f"City: {city}\n")
        f.write(f"Age: {age}\n")
        f.write(f"Hobby: {hobby}\n")
    print("Profile saved successfully!")
else:
    print("Profile not saved.")
```