Building Interactive Programs

AiCE Software Engineering Foundations — Python Programming

With input() and type conversion in hand, this lesson puts them to real use — collecting several pieces of information at once, building genuinely useful interactive calculators, and combining everything from this program so far into a complete mini-application.

1. Getting Multiple Inputs

Most programs need more than one piece of information. You simply call input() multiple times, once for each piece of data:

```
full_name = input("Full name : ")
country   = input("Country   : ")
age       = int(input("Age       : "))

# Clean the inputs
full_name = full_name.strip().title()
country   = country.strip().title()

print(f"Name    : {full_name}")
print(f"Country : {country}")
print(f"Age     : {age}")
```

This is a pattern you'll use constantly — collect several inputs, clean each one with the string methods from an earlier lesson, then display a formatted summary.

> **A Preview of Conditional Logic**
> *You may see an inline expression like {"Beginner" if years_coding < 2 else "Advanced"} inside an f-string in some examples ahead. This is a conditional expression — covered fully in the next lesson. For now, just notice that it checks a value and picks one of two labels based on the result.*

2. Building Interactive Calculators

A series of small, genuinely useful programs that combine input(), type conversion, arithmetic, and string formatting.

Program 1 — Currency Converter

```
NGN_TO_USD = 0.00063
NGN_TO_KES = 0.082

amount = float(input("Enter amount in Naira (NGN): "))

usd = amount * NGN_TO_USD
kes = amount * NGN_TO_KES

print(f"NGN {amount:,.2f} is equal to:")
print(f"USD ${usd:,.2f}")
print(f"KES {kes:,.2f}")
```

Entering 50000 converts the Naira amount into other currencies simultaneously, using the exchange rates defined at the top.

Program 2 — BMI Calculator

```
# BMI = weight(kg) / height(m) ** 2
name   = input("Enter your name : ").strip().title()
weight = float(input("Weight in kg    : "))
height = float(input("Height in metres: "))

bmi = weight / (height ** 2)

if bmi < 18.5:
    category = "Underweight"
elif bmi < 25:
    category = "Normal weight"
else:
    category = "Overweight"
```

```
print(f"BMI      : {bmi:.2f}")
print(f"Category : {category}")
```

This calculates BMI and categorizes it — the same underlying logic that runs in hospital systems, fitness apps, and nutrition platforms in production today.

Program 3 — Student Grade Calculator

```
name   = input("Student name: ").strip().title()
score1 = float(input("Score 1 (out of 100): "))
score2 = float(input("Score 2 (out of 100): "))
score3 = float(input("Score 3 (out of 100): "))

average = (score1 + score2 + score3) / 3

if average >= 70:
    grade = "A"
elif average >= 50:
    grade = "C"
else:
    grade = "F"

print(f"Average: {average:.2f}")
print(f"Grade  : {grade}")
```

This collects three scores, computes an average, and assigns a letter grade — the foundation of a larger Student Management System built later in this program.

3. Input With Format Hints in the Prompt

Giving the user a hint about the expected format, right inside the prompt itself, makes programs noticeably more user-friendly:

```
dob = input("Date of birth (DD/MM/YYYY): ")
phone = input("Phone number (+country code, e.g.
+234XXXXXXXXX): ")
```

```
# Process the date of birth
parts = dob.split("/")
day, month, year = parts[0], parts[1], parts[2]

# Clean the phone number
clean_phone = phone.replace(" ", "").replace("-", "")

print(f"Day: {day}, Month: {month}, Year: {year}")
print(f"Phone: {clean_phone}")
```

After a date like 25/06/2026 is entered, .split() from an earlier lesson divides it cleanly into day, month, and year.

4. Combining Everything — The AiCE Registration System

A complete mini-application combining user input, type conversion, string methods, arithmetic, and formatted output:

```
first_name = input("First name: ").strip().title()
last_name  = input("Last name : ").strip().title()
age        = int(input("Age       : "))
prev_gpa   = float(input("Previous GPA (0.0-4.0): "))

full_name  = first_name + " " + last_name
student_id = age * 100 + len(full_name)
gpa_percent = (prev_gpa / 4.0) * 100

if prev_gpa >= 3.0:
    status = "ADMITTED — Full Scholarship Eligible"
elif prev_gpa >= 2.0:
    status = "ADMITTED — Standard Entry"
else:
    status = "REFERRED — Please contact admissions"

print(f"Student ID : AiCE-{student_id}")
```

```
print(f"GPA        : {prev_gpa:.2f} / 4.0
({gpa_percent:.1f}%)")
print(f"Status     : {status}")
print(f"Welcome to AiCE, {first_name}!")
```

This collects several inputs, processes them with arithmetic and string operations, generates a student ID, determines an admission status, and displays a formatted confirmation — the same pattern used in real admission portals, bank account opening flows, and onboarding systems. It's built entirely from concepts covered across the last several lessons.

Key Takeaways

Real interactive programs almost always collect several inputs, clean and convert each one appropriately, and combine them with arithmetic and string formatting into a single, readable result. Currency converters, BMI calculators, and registration systems are not exotic — they're built from exactly the tools covered across this program so far. The next lesson introduces conditional statements properly — if, elif, and else — which you've already seen previewed in a few examples here, and which let your programs make real decisions based on the data they receive.