1. Variables

```
# 1. Variable 'pi' and its type
pi = 22/7
print("Value of pi:", pi)
print("Data type of pi:", type(pi))

# 2. Assigning value to 'for' (This will cause a SyntaxError because 'for' is a reserved keyword)
# for = 4

# 3. Simple Interest Calculation
principal = 1000
rate = 5
time = 3
simple_interest = (principal * rate * time) / 100
print("Simple Interest for 3 years:", simple_interest)
```

```
Value of pi: 3.142857142857143
Data type of pi: <class 'float'>
Simple Interest for 3 years: 150.0
```

2. Numbers

```
# 1. Format function with 145 and 'o' (Octal representation)
def format_number(num, fmt):
    return format(num, fmt)

result = format_number(145, 'o')
print("Formatted 145 with 'o':", result)

# 2. Pond Area and Water Calculation
radius = 84
pi_val = 3.14
pond_area = pi_val * (radius ** 2)
total_water = pond_area * 1.4
print("Pond Area:", pond_area, "sq meters")
print("Total water in pond:", int(total_water), "liters")

# 3. Speed Calculation (490m in 7 minutes)
distance = 490
time_seconds = 7 * 60
speed = distance / time_seconds
print(f"Speed:", int(speed), "m/s")
```

```
Formatted 145 with 'o': 221
Pond Area: 22155.84 sq meters
Total water in pond: 31018 liters
Speed: 1 m/s
```

3. List

```
justice_league = ["Superman", "Batman", "Wonder Woman", "Flash", "Aquaman", "Green Lantern"]

# 1. Number of members
print("1. Total Members:", len(justice_league))

# 2. Batman recruits Batgirl and Nightwing
justice_league.extend(["Batgirl", "Nightwing"])
print("2. After recruitment:", justice_league)

# 3. Wonder Woman becomes leader (Move to index 0)
justice_league.remove("Wonder Woman")
justice_league.insert(0, "Wonder Woman")
print("3. After making Wonder Woman leader:", justice_league)

# 4. Separate Aquaman and Flash using Green Lantern
justice_league.remove("Green Lantern")
flash_idx = justice_league.index("Flash")
justice_league.insert(flash_idx, "Green Lantern")
print("4. After separating Aquaman and Flash:", justice_league)

# 5. Assemble a new team
justice_league = ["Cyborg", "Shazam", "Hawkgirl", "Martian Manhunter", "Green Arrow"]
print("5. New team:", justice_league)

# 6. Sort alphabetically
justice_league.sort()
```

```
print("6. Sorted team:", justice_league)
print("Bonus: The new leader is", justice_league[0])
```

```
1. Total Members: 6
2. After recruitment: ['Superman', 'Batman', 'Wonder Woman', 'Flash', 'Aquaman', 'Green Lantern', 'Batgirl', 'Nightwing']
3. After making Wonder Woman leader: ['Wonder Woman', 'Superman', 'Batman', 'Flash', 'Aquaman', 'Green Lantern', 'Batgirl',
4. After separating Aquaman and Flash: ['Wonder Woman', 'Superman', 'Batman', 'Green Lantern', 'Flash', 'Aquaman', 'Batgirl'
5. New team: ['Cyborg', 'Shazam', 'Hawkgirl', 'Martian Manhunter', 'Green Arrow']
6. Sorted team: ['Cyborg', 'Green Arrow', 'Hawkgirl', 'Martian Manhunter', 'Shazam']
Bonus: The new leader is Cyborg
```

4. If condition

```
# 1. BMI Category
height = float(input("Enter height in meters: "))
weight = float(input("Enter weight in kilograms: "))
bmi = weight / (height ** 2)

if bmi >= 30:
  print("Obesity")
elif 25 <= bmi < 30:
  print("Overweight")
elif 18.5 <= bmi < 25:
  print("Normal")
else:
  print("Underweight")

# 2 & 3. City-Country Check
countries = {
    "Australia": ["Sydney", "Melbourne", "Brisbane", "Perth"],
    "UAE": ["Dubai", "Abu Dhabi", "Sharjah", "Ajman"],
    "India": ["Mumbai", "Bangalore", "Chennai", "Delhi"]
}

city1 = input("Enter first city: ")
city2 = input("Enter second city: ")

country1 = next((c for c, cities in countries.items() if city1 in cities), None)
country2 = next((c for c, cities in countries.items() if city2 in cities), None)

if country1 == country2 and country1:
    print("Both cities are in", country1)
else:
    print("They don't belong to the same country")
```

```
Enter height in meters: 4
Enter weight in kilograms: 90
Underweight
Enter first city: Dubai
Enter second city: Mumbai
They don't belong to the same country
```

5. For loop

```
import random

# 1. Die Roll Statistics
rolls = [random.randint(1, 6) for _ in range(20)]
sixes = rolls.count(6)
ones = rolls.count(1)
double_six = sum(1 for i in range(len(rolls)-1) if rolls[i] == 6 and rolls[i+1] == 6)

print("Rolls:", rolls)
print("Total 6s:", sixes, "Total 1s:", ones, "Two 6s in a row:", double_six)

# 2. Workout Routine
total_jacks = 0
for i in range(1, 11):
    total_jacks += 10
    print("Completed", total_jacks, "jumping jacks.")
    if total_jacks == 100:
        print("Congratulations! You completed the workout.")
        break

    tired = input("Are you tired? (y/n): ").lower()
    if tired in ['y', 'yes']:
        skip = input("Do you want to skip remaining sets? (y/n): ").lower()
        if skip in ['y', 'yes']:
            print("You completed a total of", total_jacks, "jumping jacks.")
```

```
            break
    print(100 - total_jacks, "jumping jacks remaining.")
```

```
Rolls: [6, 2, 2, 2, 1, 4, 4, 2, 6, 2, 5, 2, 4, 3, 2, 1, 4, 6, 2, 2]
Total 6s: 3 Total 1s: 2 Two 6s in a row: 0
Completed 10 jumping jacks.
Are you tired? (y/n): Yes
Do you want to skip remaining sets? (y/n): Yes
You completed a total of 10 jumping jacks.
```

6. Dictionary

```
# 1. Friends list and tuples
friends = ["Aditya", "Rahul", "Priya", "Sneha", "Amit"]
friend_tuples = [(name, len(name)) for name in friends]
print("Friend Name Lengths:", friend_tuples)

# 2. Trip Expenses
my_expenses = {"Hotel": 1200, "Food": 800, "Transportation": 500, "Attractions": 300, "Miscellaneous": 200}
partner_expenses = {"Hotel": 1000, "Food": 900, "Transportation": 600, "Attractions": 400, "Miscellaneous": 150}

my_total = sum(my_expenses.values())
partner_total = sum(partner_expenses.values())

print("My total:", my_total, "Partner total:", partner_total)
print("I spent more" if my_total > partner_total else "Partner spent more")

# Find significant difference
diff_cat = max(my_expenses, key=lambda k: abs(my_expenses[k] - partner_expenses[k]))
print("Max difference in", diff_cat, (abs(my_expenses[diff_cat] - partner_expenses[diff_cat])))
```

```
Friend Name Lengths: [('Aditya', 6), ('Rahul', 5), ('Priya', 5), ('Sneha', 5), ('Amit', 4)]
My total: 3000 Partner total: 3050
Partner spent more
Max difference in Hotel 200
```