Solution 1 – Functions & Variables: Temperature Converter

```
def convert_temperature(value, unit):
    if unit == "C":
        return (value * 9/5) + 32
    elif unit == "F":
        return (value - 32) * 5/9
    else:
        return None

# Challenge version
conversions = {
    ("C", "F"): lambda x: (x * 9/5) + 32,
    ("F", "C"): lambda x: (x - 32) * 5/9,
    ("C", "K"): lambda x: x + 273.15,
    ("K", "C"): lambda x: x - 273.15
}
```

Solution 2 – Lists & Loops: Even Number Filter

```
def filter_even(numbers):
    evens = []
    for n in numbers:
        if n % 2 == 0:
            evens.append(n)
    return evens

# Challenge version
def filter_numbers(numbers, even=True):
    return [n for n in numbers if (n % 2 == 0) == even]
```

Solution 3 – Dictionaries: Word Counter

```
def word_count(text):
    words = text.split()
    counts = {}
    for word in words:
        counts[word] = counts.get(word, 0) + 1
    return counts
```

```python
# Challenge version
from collections import Counter
import re

def word_count(text):
    words = re.findall(r"\w+", text.lower())
    return dict(Counter(words))
```

Solution 4 – Control Loops: Multiplication Table

```python
def multiplication_table(n):
    for i in range(1, n+1):
        for j in range(1, n+1):
            print(i * j, end="\t")
        print()

# Challenge version
def multiplication_table(n):
    return [[i*j for j in range(1, n+1)] for i in range(1, n+1)]
```

Solution 5 – OOP with Python

```python
import random

class Shape:
    def __init__(self):
        pass

    def area(self):
        pass

class Circle(Shape):
    def __init__(self):
        super().__init__()
        self.radius = 0

    def setRadius(self, radius):
        self.radius = radius

    def area(self):
        return 3 * self.radius * self.radius

class Square(Shape):
    def __init__(self):
```

```
        super().__init__()
        self.length = 0

    def setLength(self, length):
        self.length = length

    def area(self):
        return self.length * self.length

# Instantiate circle and square
c = Circle()
c.setRadius(5)
print("Circle area:", c.area())

s = Square()
s.setLength(4)
print("Square area:", s.area())

# Polymorphism
shapes = []
for _ in range(10):
    if random.choice([True, False]):
        circle = Circle()
        circle.setRadius(random.randint(1, 10))
        shapes.append(circle)
    else:
        square = Square()
        square.setLength(random.randint(1, 10))
        shapes.append(square)

total_area = 0
for i,shape in enumerate(shapes):
    area_val = shape.area()
    print(f"Area of shape nr. {i}:", area_val)
    total_area += area_val

print("Total area:", total_area)
```

Solution 6 – Combining Concepts: Student Grades

```
def best_student(grades_dict):
    best_name = None
    best_avg = 0
    for name, grades in grades_dict.items():
        avg = sum(grades) / len(grades)
        print(f"{name}: {avg:.2f}")
        if avg > best_avg:
            best_avg = avg
            best_name = name
```

```
    return best_name, best_avg

grades = {
    "Alice": [85, 90, 92],
    "Bob": [70, 80, 65],
    "Charlie": [95, 100, 98]
}

print("Best student:", best_student(grades))

# Challenge version
averages = {name: sum(g)/len(g) for name, g in grades.items()}
best = max(averages, key=averages.get)
print("Best student:", best, averages[best])
```