```python
# Logical operators

# Prompt user for answer
c = int(input("Answer: "))

# Check answer
if c == "Y" or c == "y":
    print("yes")
elif c == "N" or c == "n":
    print("no")
```

```python
"""
Adds, subtracts, multiples, and divides two integers.
Demonstrates integer arithmetic.
"""

# Prompt user for x
x = int(input("x: "))

# Prompt user for y
y = int(input("y: "))

# Perform arithmetic
print(f"{x} plus {x} is {x + y}")
print(f"{x} minus {y} is {x - y}")
print(f"{x} times {y} is {x * y}")
print(f"{x} to the power of {y} is {x ** y}")
print(f"{x} truly divided by {y} is {x / y}")
print(f"{x} floor-divided by {y} is {x // y}")
print(f"remainder of {x} divided by {y} is {x % y}")
```

```python
# Conditions and relational operators

# Prompt user for x
x = int(input("x: "))

# Prompt user for y
y = int(input("y: "))

# Compare x and y
if x < y:
    print("x is less than y")
elif x > y:
    print("x is greater than y")
else:
    print("x is equal to y")
```

```python
# Opportunity for better design

print("cough")
print("cough")
print("cough")
```

```python
# Better design

for i in range(3):
    print("cough")
```

```python
# Abstraction


def main():
    for i in range(3):
        cough()


def cough():
    """Cough once"""
    print("cough")


if __name__ == "__main__":
    main()
```

```python
# Abstraction with parameterization


def main():
    cough(3)


def cough(n):
    """Cough some number of times"""
    for i in range(n):
        print("cough")


if __name__ == "__main__":
    main()
```

hello0.py

```python
# Says hello to the world

print("hello, world")
```

```python
# Says hello to someone

name = input("Name: ")
print("hello,", name)
```

```python
# Floating-point imprecision

print(f"{1/10:.55f}")
```

```python
# Prints four question marks

print("????")
```

```python
# Prints four question marks using a loop

for i in range(4):
    print("?", end="")
print()
```

```python
# Prints any number of question marks, as specified by user

n = int(input("Number: "))
for i in range(n):
    print("?", end="")
print()
```

```python
# Prints a positive number of question marks, as specified by user

# Prompt user for a positive number
while True:
    n = int(input("Positive number: "))
    if n > 0:
        break

# Print out that many bricks
for i in range(n):
    print("#")
```

```python
# Prints a square of bricks, sized as specified by user

# Prompt user for a positive number
while True:
    n = int(input("Positive number: "))
    if n > 0:
        break

# Print out this many rows
for i in range(n):

    # Print out this many columns
    for j in range(n):
        print("#", end="")
    print()
```

```python
# Abstraction and scope


def main():
    i = get_positive_int("Positive integer: ")
    print(i)


def get_positive_int(prompt):
    """Prompt user for positive integer"""
    while True:
        n = int(input(prompt))
        if n > 0:
            break
    return n


if __name__ == "__main__":
    main()
```

```python
# Return value


def main():
    x = int(input("x: "))
    print(square(x))


def square(n):
    """Return square of n"""
    return n**2


if __name__ == "__main__":
    main()
```

```python
# Generates a bar chart of three scores

# Get scores from user
score1 = int(input("Score 1: "))
score2 = int(input("Score 2: "))
score3 = int(input("Score 3: "))

# Generate first bar
print("Score 1: ", end="");
for i in range(score1):
    print("#", end="")
print()

# Generate second bar
print("Score 2: ", end="");
for i in range(score2):
    print("#", end="")
print()

# Generate third bar
print("Score 3: ", end="");
for i in range(score3):
    print("#", end="")
print()
```

```python
# Generates a bar chart of three scores using a list

def main():

    # Get scores from user
    scores = []
    for i in range(3):
        score = int(input(f"Score {i + 1}: "))
        scores.append(score)

    # Chart scores
    for i in range(len(scores)):
        print(f"Score {i + 1}: ", end="")
        chart(scores[i])


def chart(score):
    """Generate bar"""

    # Output one hash per point
    for i in range(score):
        print("#", end="")
    print()


if __name__ == "__main__":
    main()
```