skillified.py

exams.py

```
def course():
    print("A Comprehensive Guide
to Basic Python Concepts.")


course()
```

TERMINAL

skillified.py

exams.py

```
# Declare list of Course Contents
course_contents = [ "Commenting in Python",
                    "Data Types",
                    "Functions",
                    "Variable Names",
                    "Lists",
                    "Modulus Operator",
                    "Adding Items to Lists",
                    "While Loops",
                    "Importing Libraries",
                    "User Input"
]
```

TERMINAL

skillified.py exams.py

```
#═══════ Comments In Python ═══════


# In Python, comments are used to explain code and make
# it more readable. Comments are ignored by the
# interpreter. The correct way to add a single-line
# comment is using the hash symbol (#).

# this is a Comment
```

TERMINAL

skillified.py exam.py

```
#═══════════ Data Types In Python ═══════════

# Python supports various data types,
# including integers (int), floating-point
# numbers (float), and strings (str).
# When performing arithmetic operations,
# the result data type depends on the
# operands.For example, the sum of an
# integer and a float is a float:


result = 5 + 3.14  # result is a float with value 8.14

```

TERMINAL

skillified.py exams.py

```python
#═══════ Creating Functions In Python ═══════


# Functions in Python are defined using the
# def keyword, followed by the function name
# and parentheses. Here's an example of a
# simple function definition:


def function_name():
    print("This is a function.")
```

TERMINAL

skillified.py

exams.py

```
#═══════ Valid Variable Names In Python ═══════


 # Variable names in Python can contain
 # letters, numbers, and underscores,
 # but they cannot start with a number or
 # contain special characters like hyphens.

 # For example:

 my_var = 10  # Valid
 my-var = 10  # Invalid


```

TERMINAL

skillified.py exams.py

```
#═══════ Creating Lists In Python ═══════


# Lists in Python are ordered collections of
# items, and they can contain elements of
# different data types.
# A list is created using square brackets.

# For example:

my_list = [1, 2, 3]


```

TERMINAL

skillified.py exams.py

```python
#═══════ The Modulus Operator In Python ═══════


# The modulus operator (%) returns the
# remainder of a division.

# For example:


remainder = 10 % 3  # remainder is 1
```

TERMINAL

skillified.py exams.py

```
#═══════ Adding Items To Lists In Python ═══════


# You can add an item to the
# end of a list using the
# append() method.

# For example:

my_list.append(4)
```

TERMINAL

skillified.py exams.py

```python
#═══════════ While Loops In Python ═══════════


# A while loop in Python executes
# a block of code as long as a
# condition is true.

# Here's an example of a while loop that
# continues until i is 5:

i = 0
while i < 5:
    print(i)
    i += 1
```

TERMINAL

skillified.py exams.py

```
#═══════ Importing Libraries In Python ═══════


# Libraries in Python are imported using the
# import keyword, followed by the library's name.
# This allows you to access the functions and classes
# provided by the library.

# For example:

import math
```

TERMINAL

skillified.py

exams.py

```
#═══════ Reading User Input In Python ═══════


# The input() function is used to read input
# from the user in Python.
# You can store the user's input in a
# variable for further processing.

# For example:

name = input("Enter your name: ")

print("Hello, " + name + "!")

```

TERMINAL

skillified.py exams.py

```
#═══════════ Course Complete ═══════════

#═══════════ Great Work! ═══════════

# Congratulations on completing the
# Introduction to Python course!
# You've learned about fundamental concepts
# such as commenting, data types, functions,
# variable names, lists, loops, and more.

# You're now ready to take the exam on the
# Streamlit application.

# Good luck!
```

TERMINAL