CSC 223 - Advanced Scientific Programming

Basic Python Semantics

Syntax and Semantics

- The *syntax* of a programming language refers to structure of the language, that is, what constitutes a legal program.
- The *semantics* of a programming language refers to the meaning of a legal program.

Variables

- A Python variable binds a name to a value.
- New bindings are established using the assignment statement:

```
# assign 4 to the variable x
x = 4
```

- Variable naming rules:
  - A variable name may include only the characters a-z, A-Z, 0-9, and the underscore
  - A variable name must start with a letter or an underscore
  - Variable names are case sensitive

Python Objects

- Every value in Python is an object
- Objects have attributes (state) and methods (behavior)
- Syntax for using a method:

  ```
  object.method([parameters])
  ```

- Example:

  ```
  >>> x = 4.5
  >>> x.is_integer()
  False
  ```

Arithmetic Operators

Expression Type	Operator	Description
Addition	`a + b`	Sum of a and b
Subtraction	`a - b`	Difference of a and b
Multiplication	`a * b`	Product of a and b
Division	`a / b`	Quotient of a and b
Floor Division	`a // b`	Quotient, removing fractional parts
Modulus	`a % b`	Remainder after division of a by b
Exponentiation	`a ** b`	a raised to the power of b

Bitwise Operators

Expression Type	Operator	Description
Bitwise AND	a & b	Bits defined in both a and b
Bitwise OR	a | b	Bits defined in a or b
Bitwise XOR	a ^ b	Bits defined in a or b, not both
Bit shift left	a << b	Shift bits of a left by b units
Bit shift right	a >> b	Shift bits of a right by b units
Bitwise NOT	~a	Bitwise negation of a

Assignment Operators

Operator	Equivalent to
a += b	a = a + b
a -= b	a = a - b
a /= b	a = a / b
a //= b	a = a // b
a %= b	a = a % b
a *= b	a = a * b
a &= b	a = a & b
a |= b	a = a | b
a ^= b	a = a ^ b
a <<= b	a = a << b
a >>= b	a = a >> b

Comparison Operators

Operator	Description
`a == b`	a equal to b
`a != b`	a not equal to b
`a < b`	a less than b
`a > b`	a greater than b
`a <= b`	a less than or equal to b
`a >= b`	a greater than or equal to b

Logical Operators

Operator	Description
`a and b`	True if both a and b are true
`a or b`	True if either a or b is true
`not a`	True if a is False

Identity and Membership Operators

Operator	Description
`a is b`	True if a and b are identical objects
`a is not b`	True if a and b are not identical objects
`a in b`	True if a is a member of b
`a not in b`	True if a is not a member of b