



ICT 133

Structured Programming

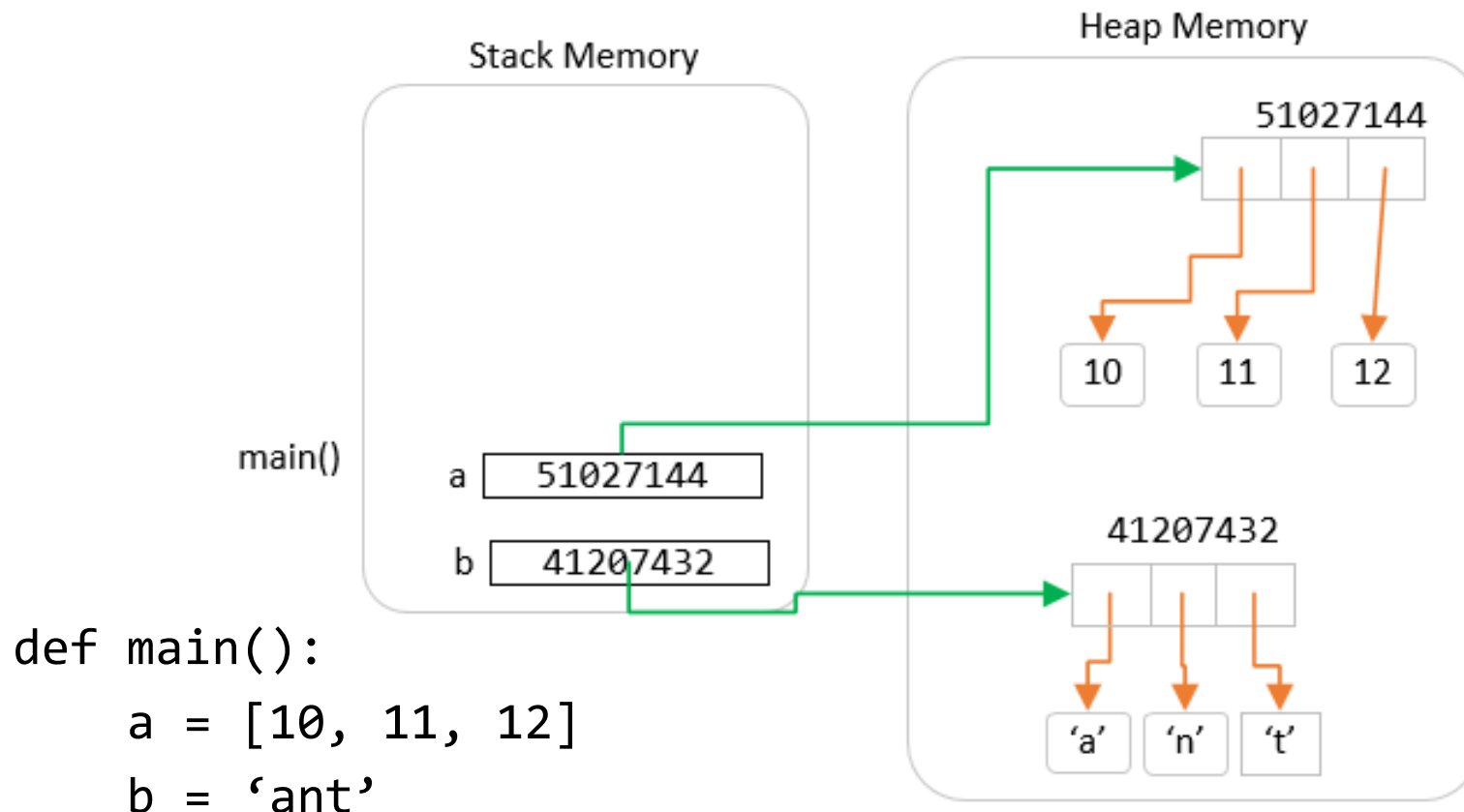
Seminar 6



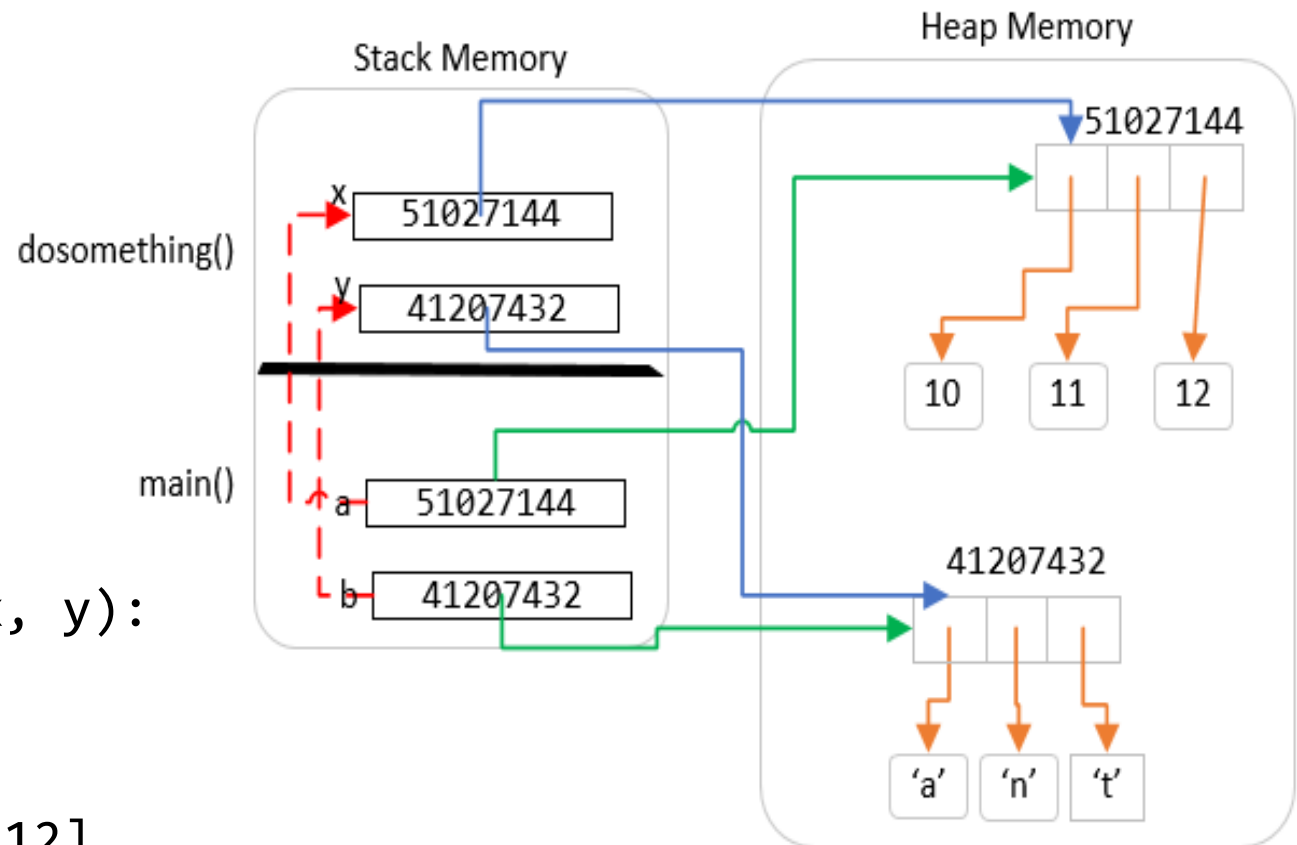
Topics

- Variables and objects
- Classes
- Python class documentation.
- Encapsulation.

Variables and objects

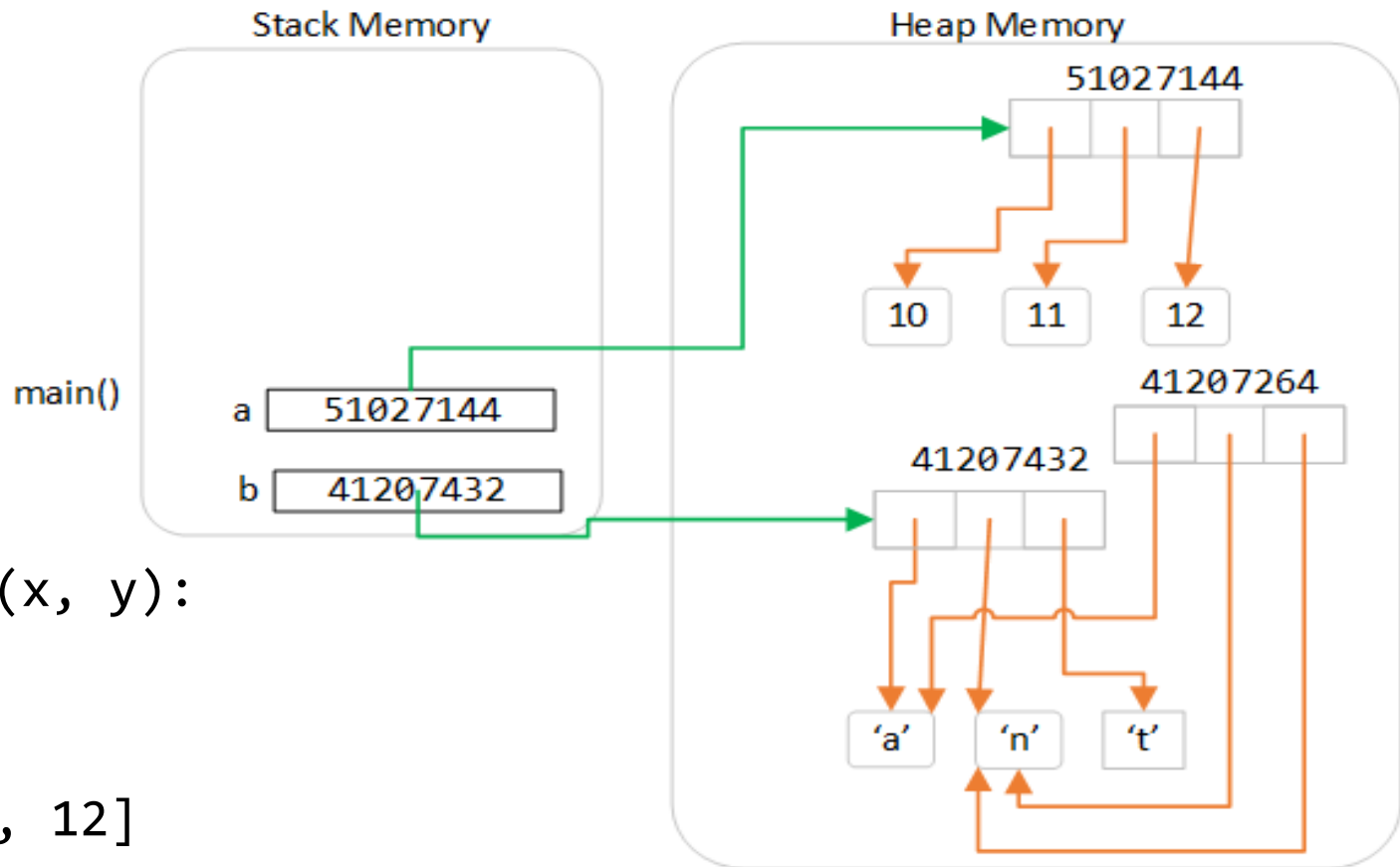


Variables and objects



```
def dosomething(x, y):  
    ...  
def main():  
    a = [10, 11, 12]  
    b = 'ant'  
    dosomething(a, b)
```

Variables and objects



```
def dosomething(x, y):  
    ...  
def main():  
    a = [10, 11, 12]  
    b = 'ant'  
    dosomething(a, b)
```



Classes

- Data type that defines the data structure and operations on the data.
 - E.g., `str` is a class.
- An object is a specific occurrence of a class.
 - E.g., `'hello'` is an object from the class `str`



Class documentation

```
>>>help(str)
```

```
Help on class str in module builtins:
```

```
class str(object)
```

```
| str(object="") -> str
```

```
| str(bytes_or_buffer[, encoding[,  
errors]]) -> str
```

```
|
```

```
| Create a new string object from the  
given object. ...
```

```
...
```

```
| Methods defined here:
```

```
|
```

```
| capitalize(...)
```

```
|     S.capitalize() -> str
```

```
|
```

```
|     Return a capitalized version of S, i.e.  
make the first character
```

```
|     have upper case and the rest lower  
case..
```

```
...
```



Encapsulation

- What objects can do, not about how they are implemented.
- Separation of concerns is known as *encapsulation*.
- The implementation details of an object are encapsulated in the class definition, which insulates the rest of the program from having to deal with them.