Objects First with Java

1 Methods and Parameters

- Objects have operations which can be invoked (Java called them methods)
- Methods may have parameters to pass additional information needed to execute
- In Java methods are not ”first class” (can’t pass them around)
- Many instances can be created from a single class
- An object have attributed: values stored in fields
- The class defines what field an object has, but each object stores its own set of values (the state of the object)

2 Basic class structure

```
public class ClassName
{
        Fields
        Constructors
        Methods
}
```

3 Fields

- Fields store values for an object
- They are also known as instance variables
- Use the inspect option to view an object’s fields
- Fields define the state of an object

4 Constructors

- Constructors initialise an object
- They have the same name as their class
- They store initial values into the fields
- They often receive external parameter values for this

```
public TicketMachine(int ticketCost)
{
        price = ticketCost;
        balance = 0;
        total = 0;
}
```

5 Accessor methods

- Methods implement the behaviour of objects
- Accessors provide information about an object
- Methods have a structure consisting of a header and a body
- The header defines the method’s signature `public int getPrice()`
- The body encloses the method’s statement

```
return price;
```

6 Mutator methods

- Have a similar method structure: header and body
- Used to mutate (i.e. change) an object's state
- Achieved through changing the value of one or more fields
  - Typically contain assignment statements
  - Typically receive parameters

```
balance=balance+amount
```

7 Printing from methods

```
System.out.println("String")
```

8 If statements

```
if(perform some test) {
        //Do these statements if the test gave a true result
}
else {
        //Do these statements if the test gave a false result
}
```

9 Local variables

Fields are one sort of variable

- They store values through the life of the object
- They are accessible throughout the class

Methods can include shorter lived variables

- They exist only as long as the method is being executed
- They are only accessible from within the method

9.1 Scope and life time

- The scope of a local variable is the block it's declared in
- The lifetime of a local variable is the time of execution of the block it's declared in

9.2 Local variable syntax

```
public int refundBalance()
{
        int amountToRefund; // note there is no visibility modifier here
        amountToRefund = balance;
        balance = 0;
        return amountToRefund;
}
```