Construction of Objects

Previously stated objects resemble cells and are created from a class or a blueprint. Also, they have attributes and behavior known as internal state defined by classes. Cells have internal processes which interact within the limitations of the plasma membrane and objects have private variables which are initialized inside the constructor methods and cannot be modified directly known as encapsulation.

The simplest way would be start out with the object and from previously it is known that an object is a block of memory, space or an entity, which can be expressed for example in Java code. The creation of a bank-account object using Java can be accomplished with the following syntax **BankAccount anAccount = new BankAccount();**. This expression can be divided into a left-hand side and a right-hand side expression. The right-hand side expression contains the **new** operator that creates the object and the **BankAccount()** with parentheses meaning it's a class and a call to the constructor or **new cell**. A constructor allows for the initialization of the object and contains instance fields, which are protected variables that hold the object contents like **currentBalance**. The constructor has the same name as the class in this case it is **BankAccount** and it has an explicit parameter named **aBalance** which is a data type **double**. Notice that **aBalance** contents is stored in the instance field **currentBalance**.

```
public BankAccount ( double aBalance){

        currentBalance = aBalance;

}
```

Classes are capitalized in Java. The left-hand side contains the **BankAccount** or **IntanceTypeOf(cell)**, which is a class and the name of the object the **anAccount** or **var cellaCell**, which can be classified as a reference. The assignment operator **=** means that the memory location of the object is stored in the reference. The expression **BankAccount anAccount = new BankAccount();** can be compared to the object **o** in the OOC. A visual object summary is shown below in Fig. 6.

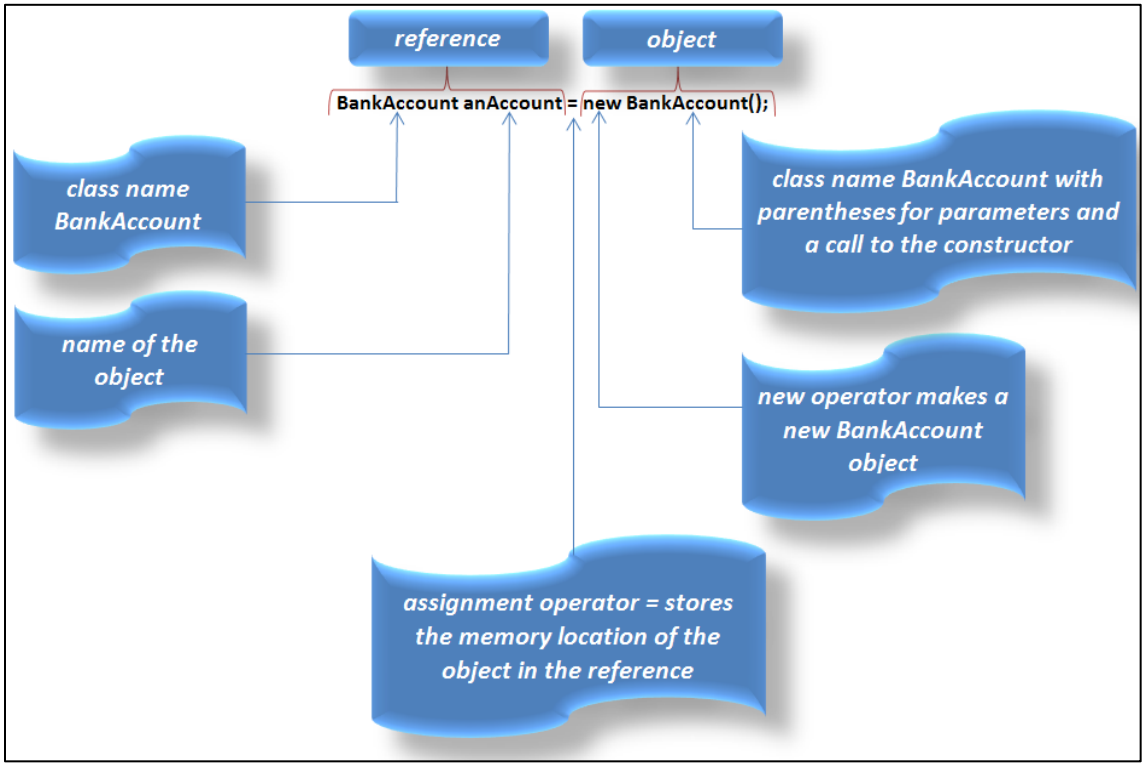

Fig. 6

Any object can be created using this structure with Java code. Again, **BankAccount anAccount = new BankAccount();** is the equivalent object **o** in OOC shown in the table below. Furthermore, this object has behavior and belongs to the class **BankAccount**.

Java Code, OOL	Procedural Language, OOC
BankAccount anAccount = new BankAccount();	**object o**

Self-Check Questions for The Construction of Objects Sub-Section

1. What is an object?
2. What does the new operator do?
3. What is a reference?
4. What is a constructor?
5. Where does the object store its contents?
6. What is an instance field?
7. What does an object have?