2.1 Objects

When humans experience new events, they can be classified as new memories or objects (cells in OOB). The objects are stored in our mind and are also known as the mental space. The mental space can be considered as HMS, which consists partially out of the WM and the LTM and is responsible for producing the codec process meaning it encodes new objects from the WM into the LTM and decodes stored objects from the LTM into the WM. The formation of new memories can be compared to creation of new objects in an OOL, such as Java where the new operator creates the object in the computer physical memory also known as a block of memory or storage space.

Objects resemble cells, which are created from a class that acts as a blueprint. Also, objects have attributes and behavior that make up the unique internal state, which are defined by classes. Just like the cells have internal processes (inside the nucleus), which interact within the boundaries of the plasma membrane or cell wall, objects have private variables which are initialized inside the constructor methods (nucleus) and cannot be modified directly. In both cases it is known as encapsulation or data security.

An object is created from a class cell using the construction **new cell**. The new cell assigns an attribute record and returns a reference or position to it. The attribute record holds the initial values and the methods code specified by the cell. Once the new cell is executed the cell object is generated and a reference to the attribute record is created. It can be classified as an object of class cell or an instance of class cell.

The creation of the object cell can be achieved by using the following notation:

```
var cellaCell: InstanceTypeOf(cell) := new cell;
```

The expression can be divided into three parts such that the first part contains the variable var called cellaCell or object name. The second part has the **InstanceTypeOf(cell)** is the type of object of class cell and the third part entails the new operator and the cell class. Part two is associated to part three by using the **:=** meaning equal by definition and part two is connected to part one by **:** meaning true to that.

Another way of writing the same expression can be done in the following way:

```
var cellaCell:cell := new cell;
```

The cell still has the same meaning as the **InstanceTypeOf(cell)** and that is the type of the object created from the cell.

2.2 Self-Check Questions for the Objects Sub-Section

1. New events can be classified as what?
2. What parts does HMS consist of?
3. What does a codec process do?
4. What can formation of new memories be compared to?
5. What does the new operator do?
6. What is the unique internal state of an object made of?
7. Cells have internal processes, which interact within the boundaries of what?
8. Where are the object private variables initialized?
9. What are the three parts that an object is made of?
10. Objects resemble what?
11. What is an object?