KV4000- Multiple Constructors

Constructors set the initial date for objects of that type. The constructor name is always exactly the same as that of the class it represents. If there is a need to create objects on the basis of different sets of data being available when we create them, then we can create multiple constructors to represent these deferent possibilities. We can have as many constructors as we need - as long as their ***signature*** [[1]](#footnote-1)is different from any others.

If we don’t specify any constructors Java creates what is known as the ***default constructor*** in the background. It will be used to create an object but it will do nothing else. If you specify one or more constructors, then this will not be created.

If we want to create Customer objects, then we call a constructor of type Customer. With Rectangle there was only one constructor which took no parameters:

**public Rectangle()**

**{**

**length = 0;**

**breadth = 0;**

**}**

All rectangle objects will be created the same - with a length & breadth of 0.

A constructor that has no parameters is known as a ‘no-args’ (no arguments) constructor.

If we want to create objects where we can set one or more attributes to specific values when we create them then we can write a constructor that takes one or more parameters:

**public TicketMachine(int ticketCost)**

**{**

**price = ticketCost;**

**balance = 0;**

**total = 0;**

**}**

The TicketMachine constructor allows the user to set the price of tickets when the object is created. If, for example, we had wanted to create ticketMachine objects where we could set both the ticket cost and have an amount of money added to the opening total (say to provide change if necessary) then we could have written the constructor as:

**public TicketMachine(int ticketCost, int openingTotal)**

**{**

**price = ticketCost;**

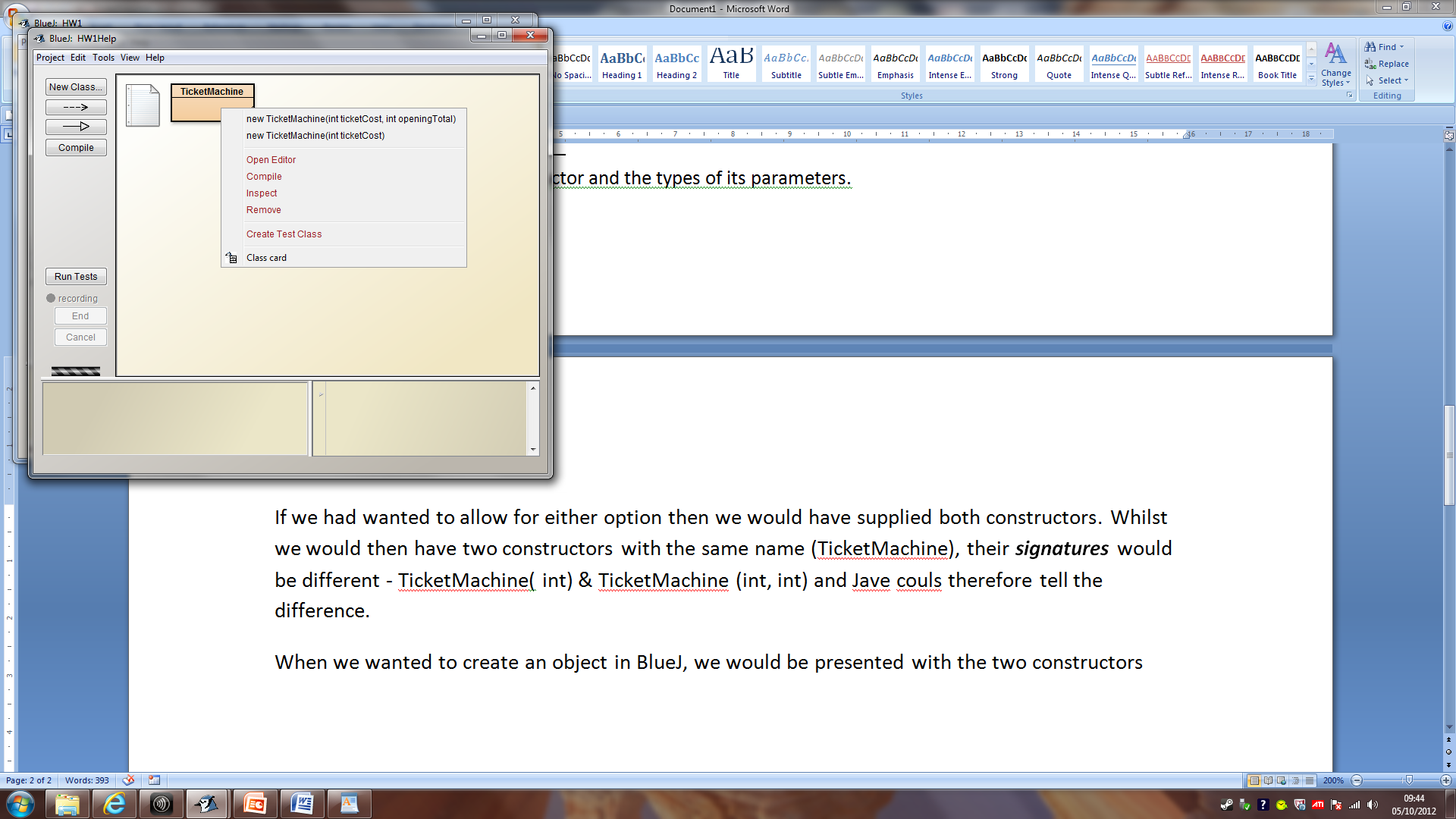
**balance = 0;**

**total = openingTotal;**

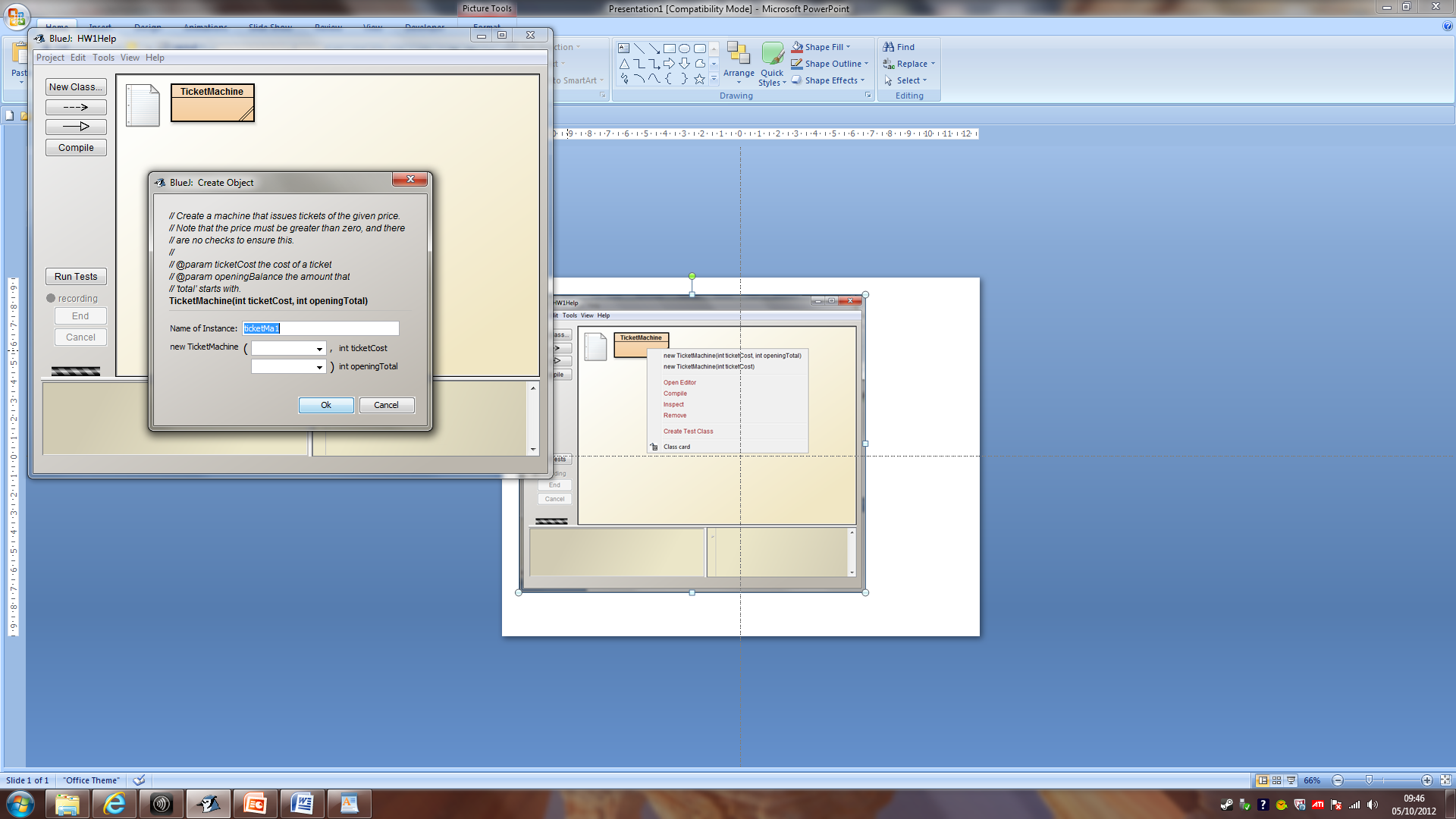
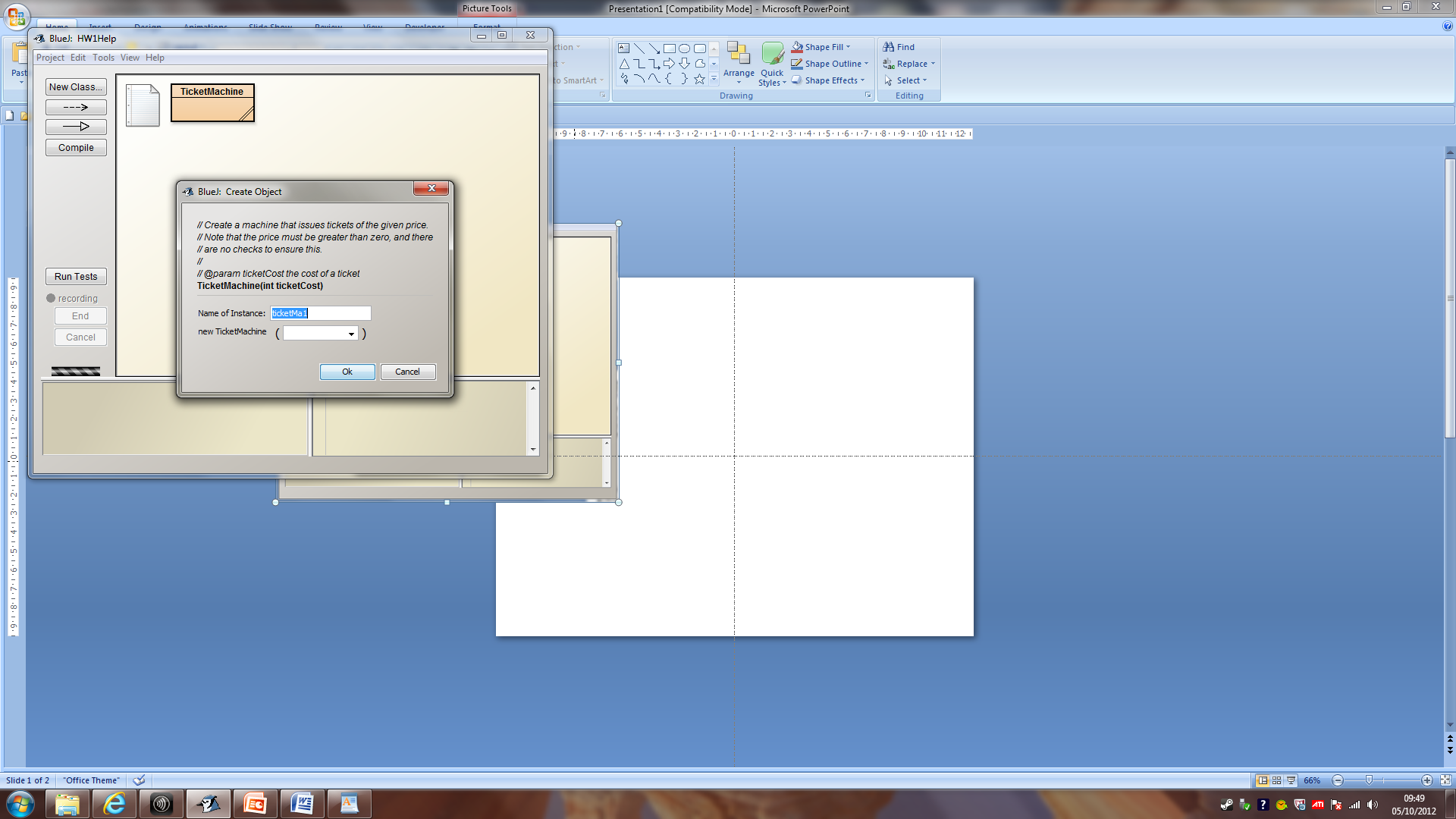
**}**

If we had wanted to allow for either option then we would have supplied both constructors. Whilst we would then have two constructors with the same name (TicketMachine), their ***signatures*** would be different - TicketMachine( int) & TicketMachine (int, int) and Jave couls therefore tell the difference.

When we wanted to create an object in BlueJ, we would be presented with the two constructors:



From there we select the one we want, enter the required values and create our object.

Suppose we wanted to create rectangles that initially have are have side of 0 as well as those where we can set the length / breadth. We would create two constructors:

|  |  |
| --- | --- |
| **public Rectangle ()**  **{**  **length = 0;**  **breadth = 0;**  **}** | **public Rectangle (int aLength, int aBreadth)**  **{**  **length = aLength;**  **breadth = aBreadth;**  **}** |

1. The name of a method / constructor and the types of its parameters. [↑](#footnote-ref-1)