Java Memory Exercise

The **Java Visualizer** allows you to run Java code line by line and see what is happening on the stack and heap.

- The Java code is shown on the left, in the green section
- Next to it, in the red section, the stack is shown
- Next to that (in purple), the heap is shown
- At the bottom (in brown) the controls

 Java Visualizer

```
</div>
```

In the example above one frame is on the stack, `main`. Within that frame are the variables created in the main method. The 'heap' is empty because no objects have been created.

Exercise 1

1. Copy and paste the code below in to the visualiser. It is the exponential code from the lecture.

```
public class Recursion {

   public static int exponential(int i)
   {
      int total;
      if (i==1)
         total = 1;
      else
      {
         total = (i * exponential(i-1));
      }
      return total;
   }

   public static void main(String[] args) {

      int iStart = 5;
      int iStartExp = exponential(iStart);

   }
}
```

2. Step through the code. Do you understand what is happening on the frame?

Exercise 2

1. Copy and paste the code below in to the visualiser. It is modified version exponential code from the lecture. This version creates an object to calculate the exponential.

```
public class MethodCalling {
   public static void main(String[] args) {

     StdOut.println("(4 + 5) - 6 = " + subtract(add(4,5),6));

    // StdOut.println("((4 + 5)/3) - (6*2) = " +
subtract(divide(add(4,5),3),multiply(6,2)));

   }

  public static int add(int num1, int num2){
    return num1 + num2;
  }

  public static int subtract(int num1, int num2){
    return num1 - num2;
  }

}
```

2. See how the `add` method is called and resolved before the `subtract` method.
3. Add in `multiply` and `divide` methods so you can uncomment the second sum. Check the code runs properly and step through it observing what happens.

Exercise 3

1. Copy and paste the code below in to the visualiser. It is modified version exponential code from the lecture. This version creates an object to calculate the exponential.

```
public class Strings {
   public static void main(String[] args) {
      String a = "Hello, world!";
      String b = "Hello, world!!".substring(0, 13);
      String c = "Hello, ";
      c += "world!";
      String d = "Hello, w"+"orld!"; // constant expr, interned
      String e = a.substring(0, 13);
      System.out.println((a == b) + " " + a.equals(b));
      System.out.println((a == c) + " " + a.equals(c));
      System.out.println((a == d) + " " + a.equals(d));
      System.out.println((a == e) + " " + a.equals(e));
   }
}
```

2. Step through the code. Do you understand what is happening? What is the difference between `a == b` and `a.equals(b)` ?

Exercise 4

1. Copy and paste the code below in to the visualiser. It is modified version exponential code from the lecture. This version creates an object to calculate the exponential.

```
public class RecursiveClass {


   public static void main(String[] args) {

      int iStart = 5;
      Exponential expObj = new Exponential(3);
      int iStartExp = expObj.exp(iStart);




   }
}

class Exponential {

   int aClassVariable = 2;

   public Exponential(int iVal)
   {
     aClassVariable = iVal;
   }

   public int exp(int i)
   {
      int total;
      if (i==1)
         total = 1;
      else
      {
         total = (i * exp(i-1));
      }
      return total;
   }
}
```

2. Step through the code. Do you understand what is happening? The object is created on the heap (on the right hand side) but the methods called from the object still cause frames to be created on the stack.