Continuation of arrays…

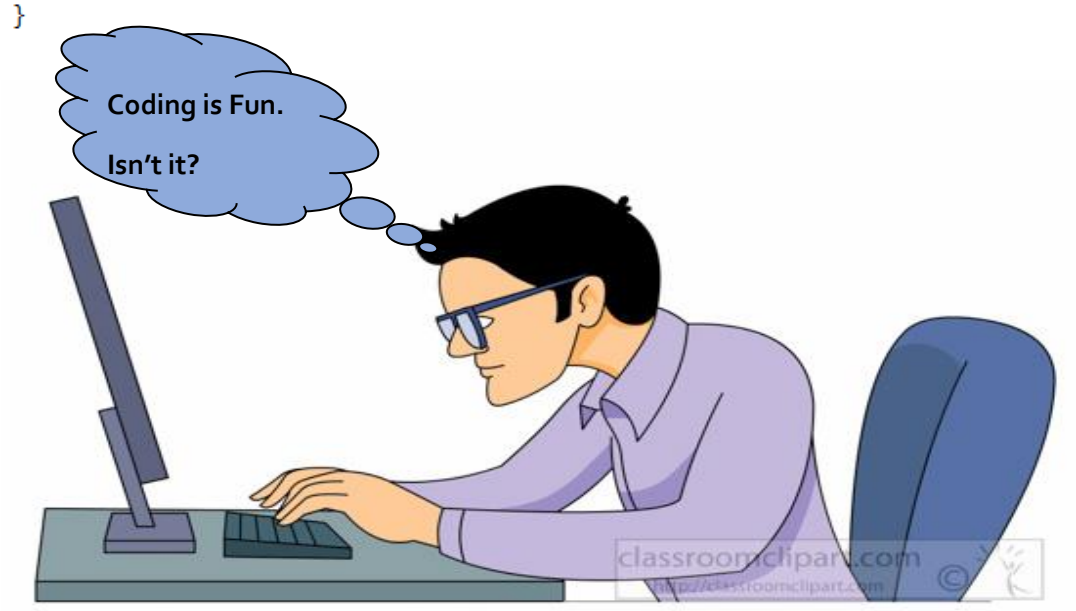
Let us try to write the code for few other examples.

Example 1): Create an array to store the ages of students belonging to 2 classrooms with 5 students each.

Solution:

```
import java.util.Scanner;
class Demo
{
    public static void main(String[] args)
    {
        int a[][] = new int[2][5];
        Scanner scan = new Scanner(System.in);

        for(int i=0;i<=a.length-1;i++)
        {
            for(int j=0;j<=a[i].length-1;j++)
            {
                    System.out.println("Enter the ages of school"+i+"class"+j+"student: ");
                    a[i][j] = scan.nextInt();
            }
        }
        System.out.println("The ages are: ");
        for(int i=0;i<=a.length-1;i++)
        {
            for(int j=0;j<=a[i].length-1;j++)
            {
                    System.out.println(a[i][j]);
            }
        }
    }
}
```

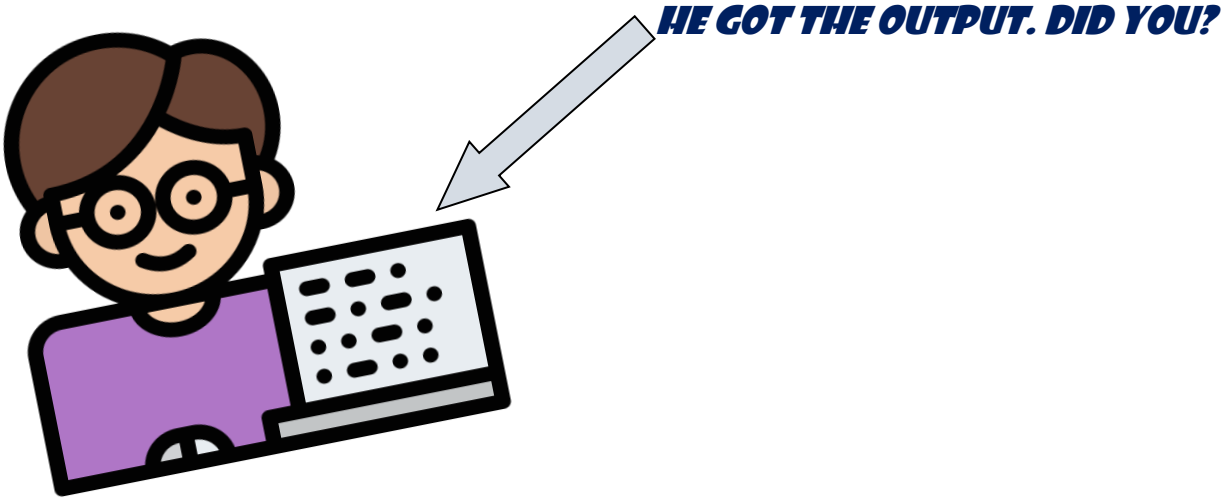


Example 2): Create an array to store the ages of students belonging to 2 schools having 3 classrooms with 5 students each.

Solution:

```
import java.util.Scanner;
class Demo
{
    public static void main(String[] args)
    {
        int a[][][] = new int[2][3][5];
        Scanner scan = new Scanner(System.in);

        for(int i=0;i<=a.length-1;i++)
        {
            for(int j=0;j<=a[i].length-1;j++)
            {
                for(int k=0;k<=a[i][j].length-1;k++)
                {
                    System.out.println("Enter the ages of school "+i+"class "+j+"student"+k);
                    a[i][j][k]= scan.nextInt();
                }
            }
        }
        System.out.println("The ages are: ");
        for(int i=0;i<=a.length-1;i++)
        {
            for(int j=0;j<=a[i].length-1;j++)
            {
                for(int k=0;k<=a[i][j].length-1;k++)
                {
                    System.out.println(a[i][j][k]);
                }
            }
        }
    }
}
```



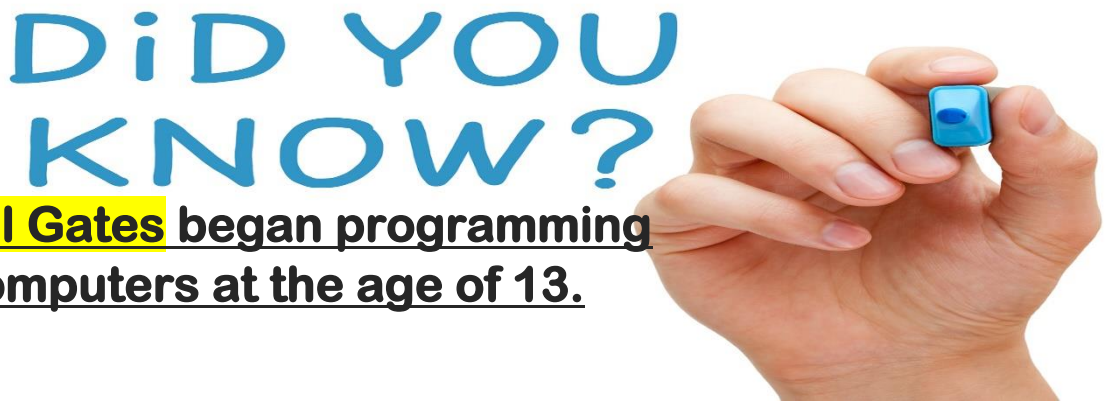
Let us look at an example of two-dimensional jagged array.

Example 3): Create an array to store the ages of students belonging to 2 classrooms where the first classroom has 3 students and second classroom has 5 students.

Solution:

```
import java.util.Scanner;
class Demo
{
    public static void main(String[] args)
    {
        int a[][] = new int[2][];
        a[0] = new int[3];
        a[1] = new int[5];
        Scanner scan = new Scanner(System.in);

        for(int i=0;i<=a.length-1;i++)
        {
            for(int j=0;j<=a[i].length-1;j++)
            {
                    System.out.println("Enter the ages of class "+i+"student"+j);
                    a[i][j]= scan.nextInt();
            }
        }
        System.out.println("The ages are: ");
        for(int i=0;i<=a.length-1;i++)
        {
            for(int j=0;j<=a[i].length-1;j++)
            {
                    System.out.println(a[i][j]);
            }
        }
    }
}
```



Bill Gates began programming Computers at the age of 13.

Similarly let us try coding for 3D jagged array.

Example 3): Create an array to store the data given below:

School	Classrooms	Students
0	0	0-1
	1	0-2
	2	0-2
1	0	0-1
	1	0-2

Solution:

```
import java.util.Scanner;
class Demo
{
    public static void main(String[] args)
    {
        int a[][][] = new int[2][][];
        a[0] = new int[3][];
        a[1] = new int[2][];
        a[0][0] = new int[2];
        a[0][1] = new int[3];
        a[0][2] = new int[3];
        a[1][0] = new int[2];
        a[1][1] = new int[3];
        Scanner scan = new Scanner(System.in);

        for(int i=0;i<=a.length-1;i++)
        {
            for(int j=0;j<=a[i].length-1;j++)
            {
                for(int k=0;k<=a[i][j].length-1;k++)
                {
                    System.out.println("Enter the ages of school "+i+"class "+j+"student"+k);
                    a[i][j][k]= scan.nextInt();
                }
            }
        }
          System.out.println("The ages are: ");
          for(int i=0;i<=a.length-1;i++)
          {
              for(int j=0;j<=a[i].length-1;j++)
              {
                  for(int k=0;k<=a[i][j].length-1;k++)
                  {
                      System.out.println(a[i][j][k]);
                  }
              }
          }
    }
}
```

Disadvantages of Array

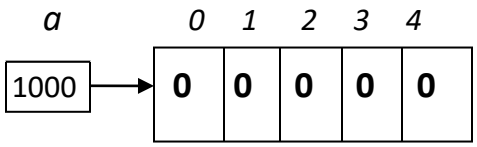
It is important for one to understand when to use an array and when not to use an array.
And to understand that, let us have a look at few Cases on Array.

Case-1:

Let us consider a one-dimensional array,

int a[ ] = new int [5];

which will be represented in the memory as,



In the above statement, we tell JVM to create an array of size 5 which can store only integer type values. Once this information is given to the JVM, what type of data it has to store, then only those types of data are allowed to store within the array.

Arrays once created can only store data of same type.

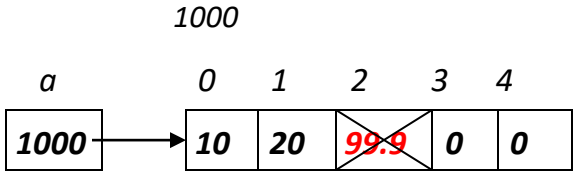
Look at the below example:

int a[ ] = new int [5];

a[0] = 10;

a[1] = 20;

a[2] = 99.9; ← ***Error***



Here, in the example in a[2] = 99.9; we are trying to store float type of data which is not possible.

Arrays can only store homogeneous data.

Case-2:

Let us consider a one-dimensional array,

int a[ ] = new int [5];

a[0] = 10;

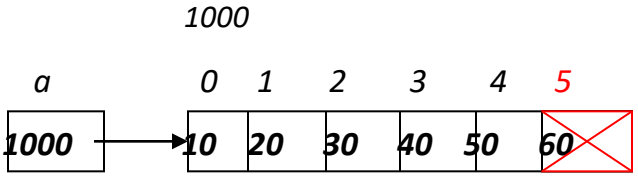
a[1] = 20;

a[2] = 30;

a[3] = 40;

a[4] = 50;

a[5] = 60;



In the above example, we have created an array of size 5. But by assigning ***a[5] = 60;***

We are trying to store value 60 at the index 5, which means adding an extra size to the array. But this is not possible because the size of array is fixed, it cannot grow or shrink in size.

Arrays cannot grow or shrink in size.

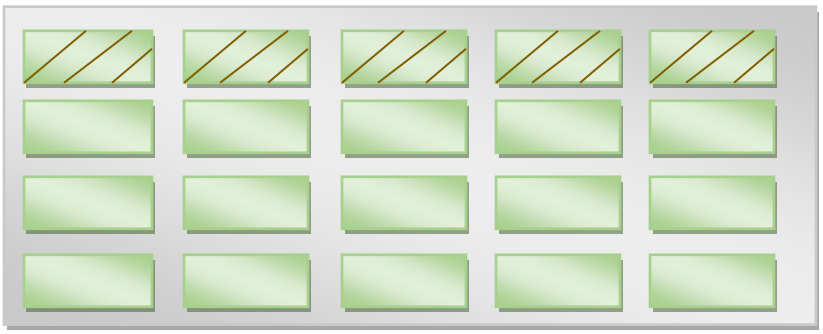
Case-3:

We know that RAM is the main memory and it is collection of bytes, if an array is created even that makes use of RAM.

Let's assume we create an array of size 5.

int a[ ] = new int [5];

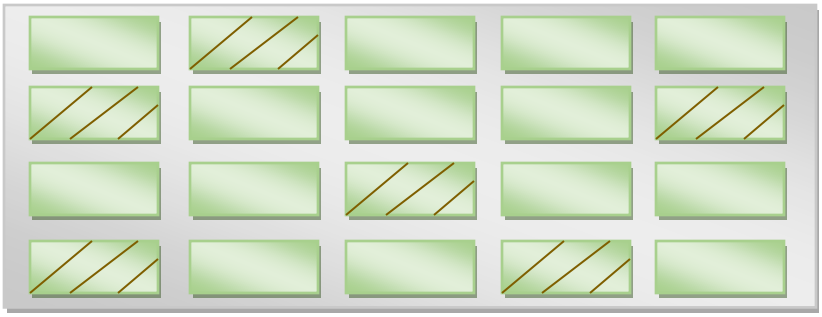
Arrays have an expectation that the size we mention to an array in terms of memory, those 5 memory locations must and should be ***contiguous*** *(which means one next to the other) as shown below.*



Contiguous memory

But in real life, in our computer multiple software's would be working and all make use of RAM. Therefore, free memory locations in reality are rarely available in RAM in continuous way, most of time it is dispersed.

Arrays demand continuous memory allocations and cannot make use of ***dispersed/scattered memory allocations****.*



Dispersed memory

Arrays require contiguous memory allocation.

Points to remember:

- **Arrays are objects**
- **They can even hold the reference variables of other objects**
- **They are created during runtime**
- **They are dynamic, created on the heap**
- **The Array length is fixed**
- **Array requires contiguous memory allocation**
- **They can only store homogeneous data.**

The original name for Java was

Oak.