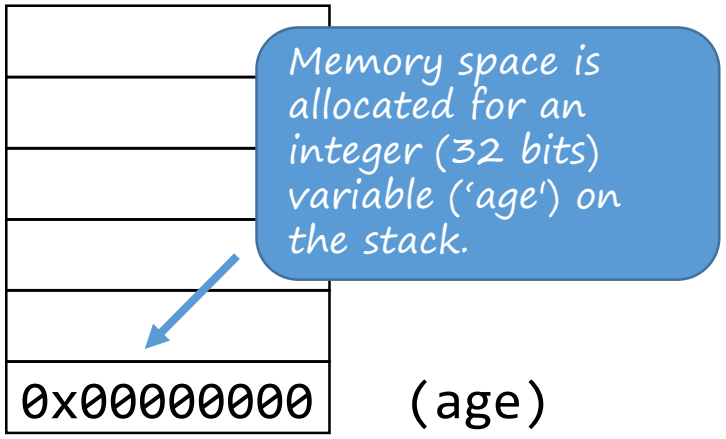
Arrays on stack/heap

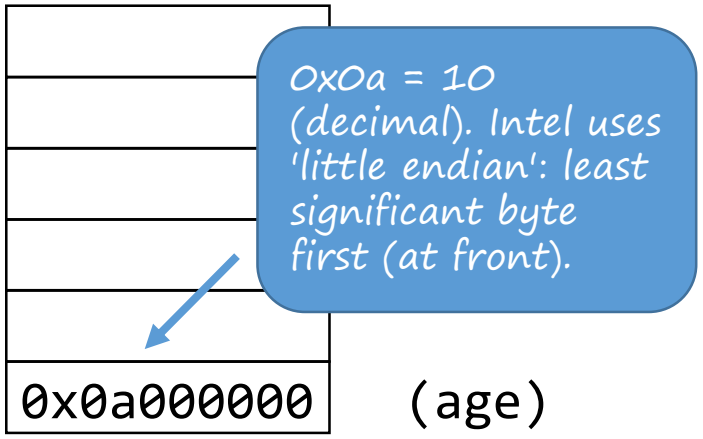
Stack
(local variables)

Heap
(objects & arrays)




```
static void Main(string[] args)
{
    int age;
}
```

Stack
(local variables)



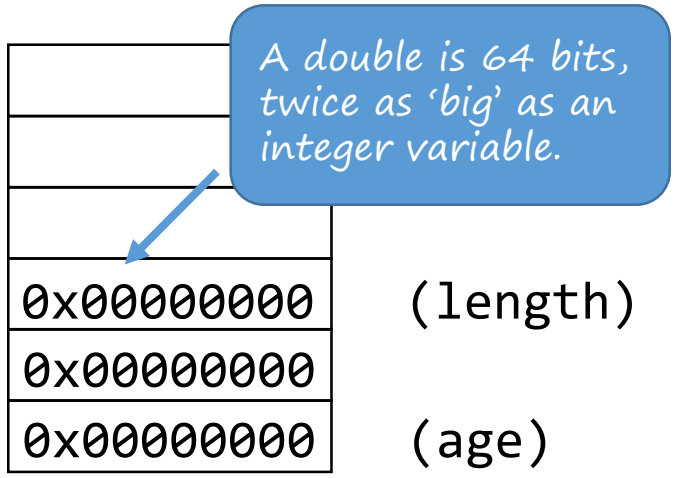
0xBE8A0D40 | 0x0a000000 | (age)

Heap
(objects & arrays)

```
static void Main(string[] args)
{
    int age;

    age = 10;
}
```

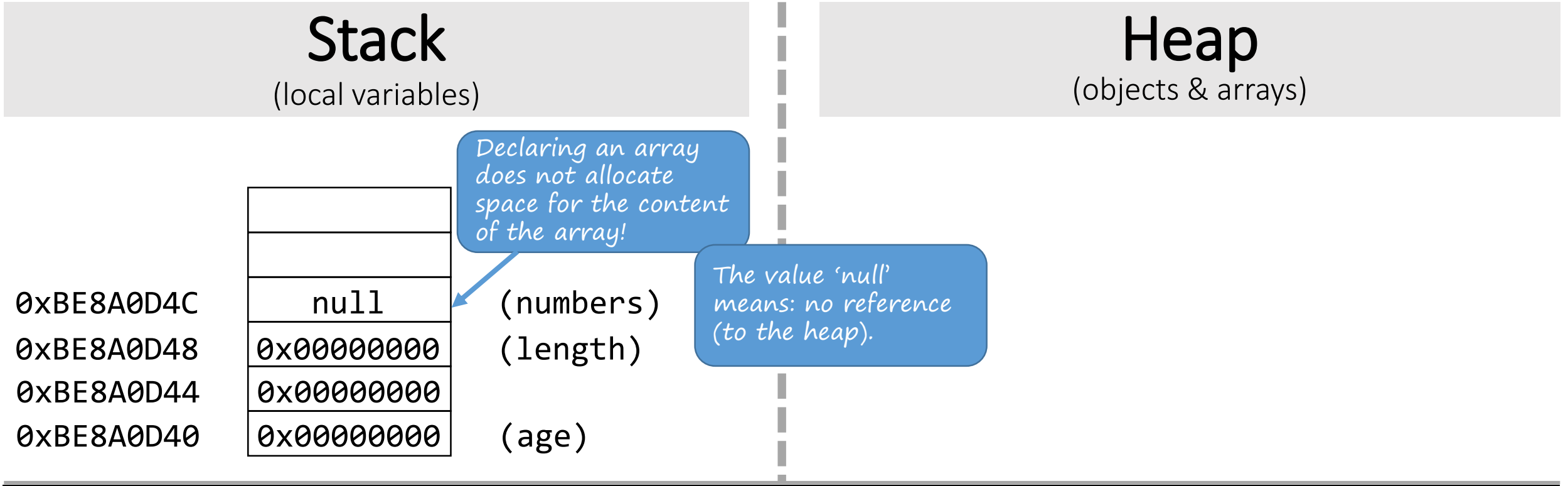
Stack
(local variables)



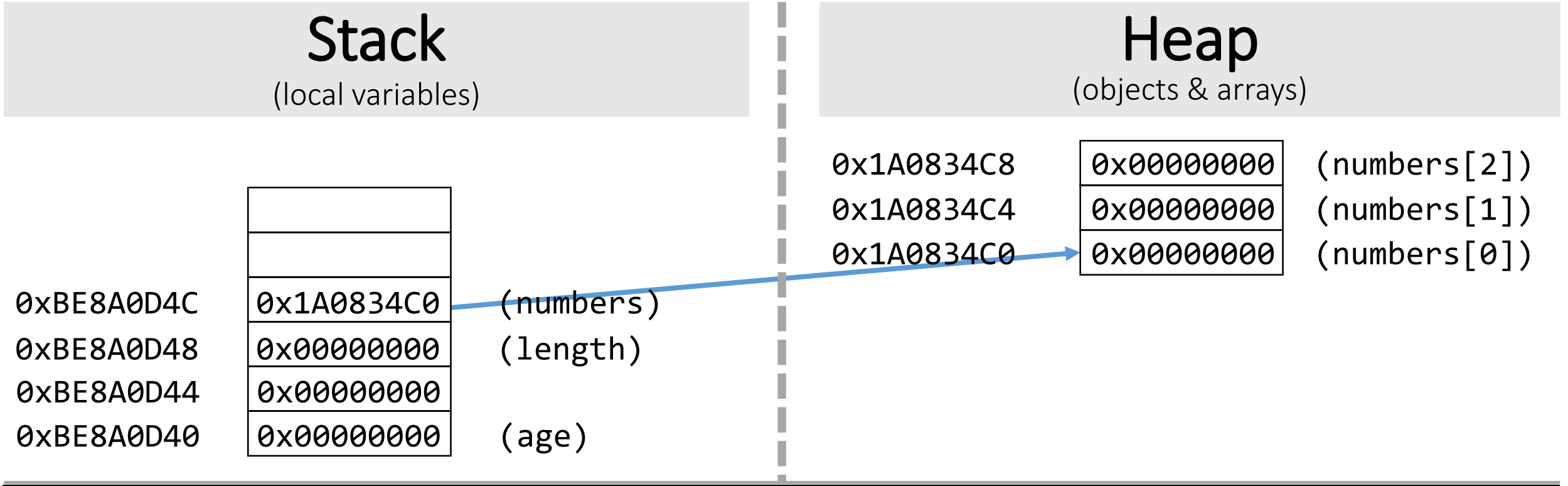
Heap
(objects & arrays)

```
static void Main(string[] args)
{
    int age;
    double length;
}
```

And what about arrays on stack?




```
static void Main(string[] args)
{
    int age;
    double length;
    int[] numbers;
}
```



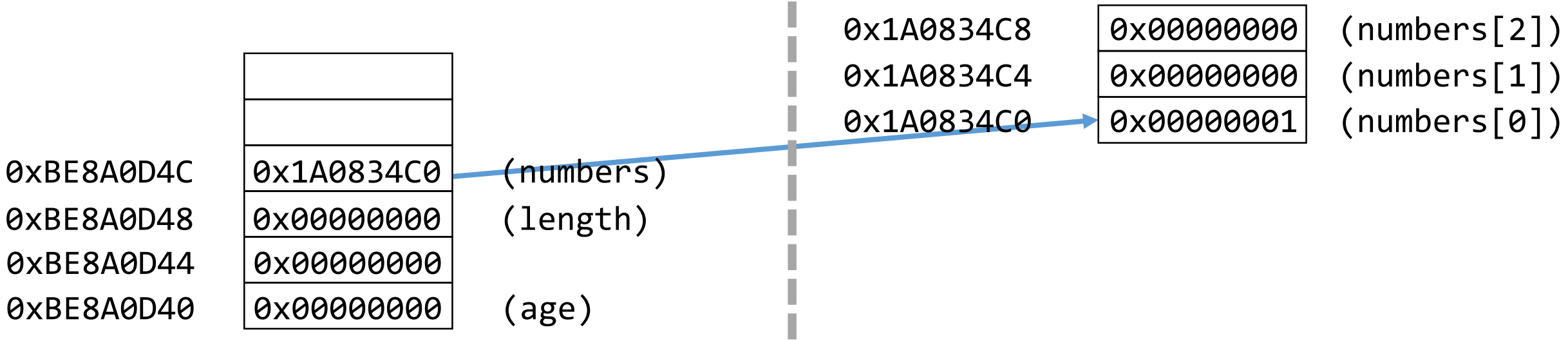

```
static void Main(string[] args)
{
    int age;
    double length;
    int[] numbers;

    numbers = new int[3];
    numbers[0] = 1;
    numbers[1] = 2;
    numbers[2] = 3;
}
```

Not until 'new ...' memory space will be allocated for the array elements.

Stack
(local variables)

Heap
(objects & arrays)

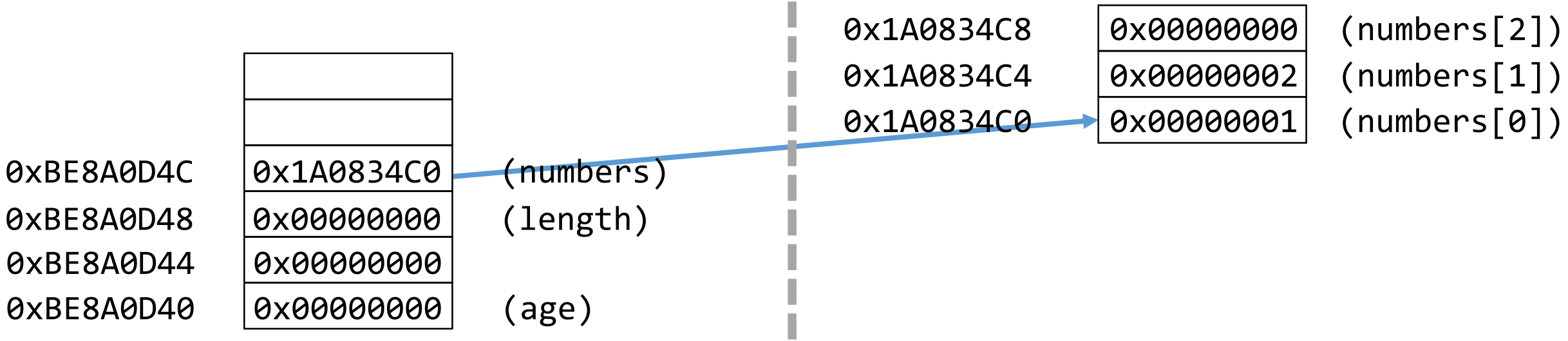



```
static void Main(string[] args)
{
    int age;
    double length;
    int[] numbers;

    numbers = new int[3];
    numbers[0] = 1;
    numbers[1] = 2;
    numbers[2] = 3;
}
```

Stack
(local variables)

Heap
(objects & arrays)

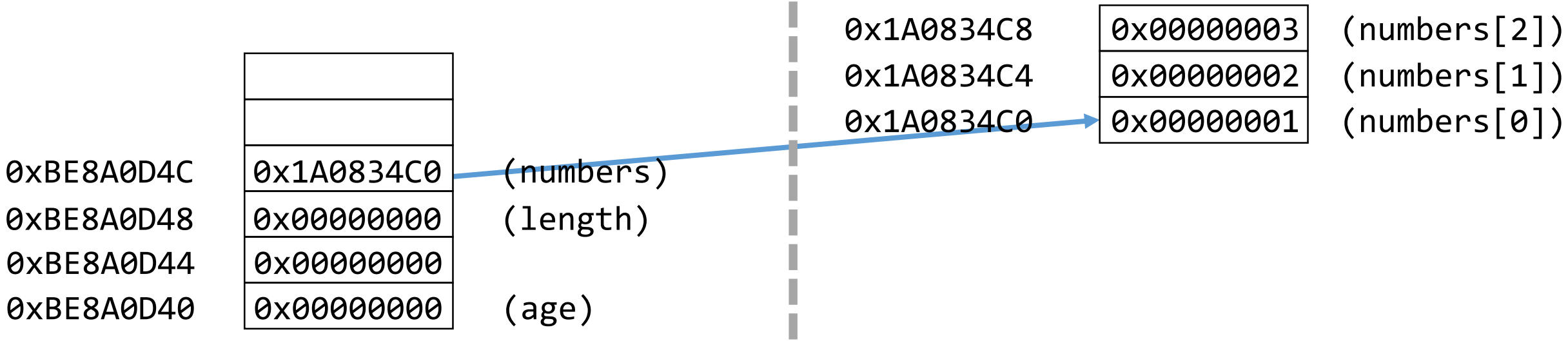



```
static void Main(string[] args)
{
    int age;
    double length;
    int[] numbers;

    numbers = new int[3];
    numbers[0] = 1;
    numbers[1] = 2;
    numbers[2] = 3;
}
```

Stack
(local variables)

Heap
(objects & arrays)




```
static void Main(string[] args)
{
    int age;
    double length;
    int[] numbers;

    numbers = new int[3];
    numbers[0] = 1;
    numbers[1] = 2;
    numbers[2] = 3;
}
```