

Pointers and functions: call-by-reference

Computer Programming 143 – Lecture 18

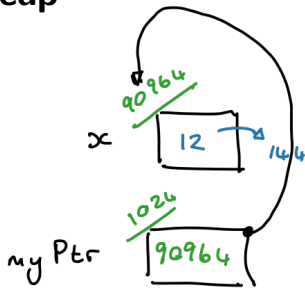
Pointers II

Faculty of Engineering

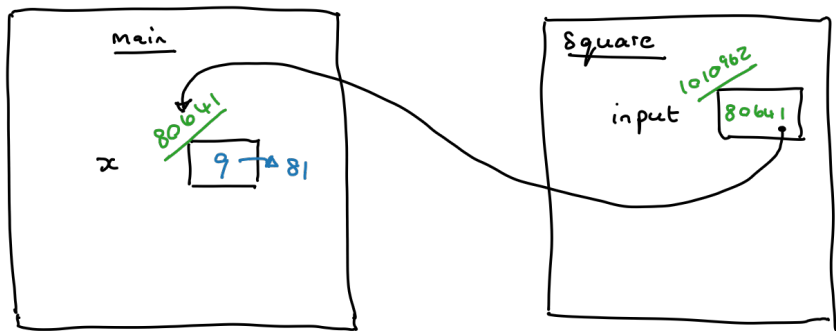


Pointer recap

- `&`: address of box, e.g. `&x`
- `*`: box that I am pointing at, e.g. `*myPtr`



Call-by-reference / pass-by-reference



Visualisation of Call-by-value

```
int main( void )
```

```
{
```

```
    int number = 5;
```

```
    number = cubeByValue( number );
```

```
}
```

number

5

```
int cubeByValue( int n )
```

```
{
```

```
    return n * n * n;
```

```
}
```

n

undefined

Visualisation of Call-by-value

```
int main( void )  
{  
    int number = 5;  
  
    number = cubeByValue( number );  
}
```

number



5

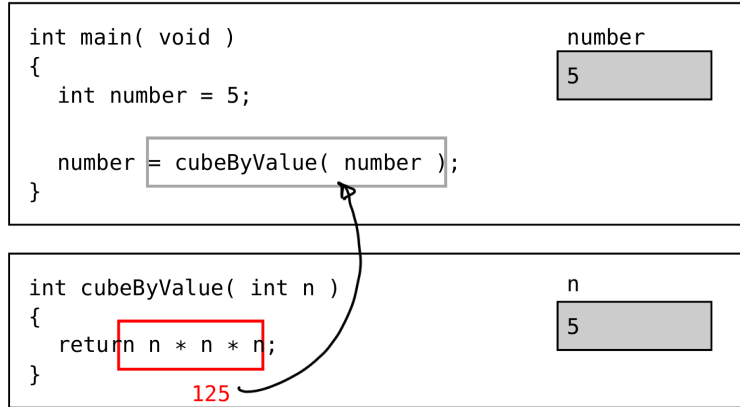
```
int cubeByValue( int n )  
{  
    return n * n * n;  
}
```

n



5

Visualisation of Call-by-value



Visualisation of Call-by-value

```
int main( void )  
{  
    int number = 5;  
  
    number = cubeByValue( number );  
}
```

125

number

5

```
int cubeByValue( int n )  
{  
    return n * n * n;  
}
```

n

undefined

Visualisation of Call-by-value

```
int main( void )
```

```
{
```

```
    int number = 5;
```

```
}
```

```
    number = cubeByValue( number );
```

125

number

125

```
int cubeByValue( int n )
```

```
{
```

```
    return n * n * n;
```

```
}
```

n

undefined

Visualisation of Call-by-reference

```
int main( void )  
{  
    int number = 5;  
  
    cubeByReference( &number );  
}
```

number

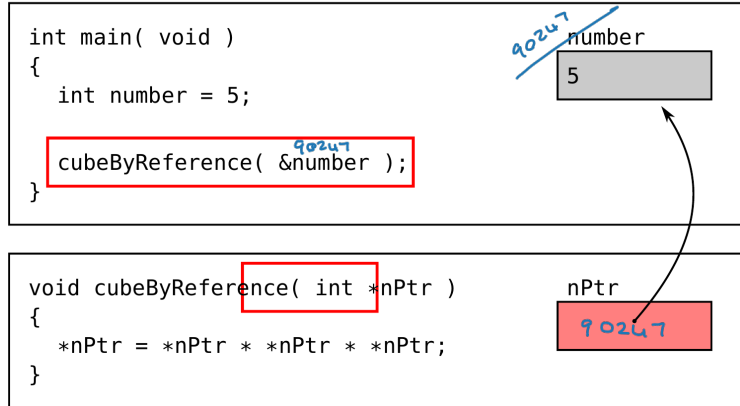
5

```
void cubeByReference( int *nPtr )  
{  
    *nPtr = *nPtr * *nPtr * *nPtr;  
}
```

nPtr

undefined

Visualisation of Call-by-reference



Visualisation of Call-by-reference

```
int main( void )  
{  
    int number = 5;  
  
    cubeByReference( &number );  
}
```

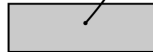
number



5 → 125

```
void cubeByReference( int *nPtr )  
{  
    *nPtr = *nPtr * *nPtr * *nPtr;  
}
```

nPtr



125

Visualisation of Call-by-reference

```
int main( void )  
{  
    int number = 5;  
    cubeByReference( &number );  
}
```

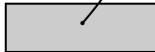
number



A red rectangular box containing the number 125, representing the memory location of the variable 'number'.

```
void cubeByReference( int *nPtr )  
{  
    *nPtr = *nPtr * *nPtr * *nPtr;  
}
```

nPtr



A grey rectangular box representing the memory location of the pointer variable 'nPtr'. An arrow points from this box to the 'number' box above.

125

Visualisation of Call-by-reference

```
int main( void )  
{  
    int number = 5;  
  
    cubeByReference( &number );  
}
```

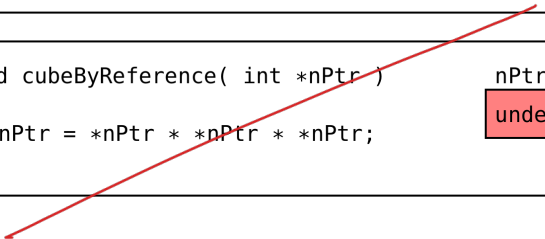
number

125

```
void cubeByReference( int *nPtr )  
{  
    *nPtr = *nPtr * *nPtr * *nPtr;  
}
```

nPtr

undefined



Swap example

