Working with Data in C++

- By default, most types in C++ will be treated

  as if they were direct values.
  - The following code will handle both variables "by value."

```
int i = 4;
Person p("Harrison Ford", 75);
```

    - Note the form of the constructor call here – it's for a "by value" class instance.

Working with Data in C++

- By default, most types in C++ will be treated

  as if they were direct values.
  - Noteworthy exception: arrays are automatically

    pointers to the actual storage.
  - The following code is valid:

```
int i[] = {1, 2, 3};
int* iArr = i; //Arrays ARE ptrs.
```

Working with Data in C++

- Conversely, any type in C++ can be referred to via a pointer.
  - The following code will handle both variables "by reference."

```
int *i = new int(4);
Person *p =
 new Person("Harrison Ford", 75);
```

    - The '*' symbol denotes that the variable stores a *pointer* to that type.

Working with Data in C++

- If a pointer is not presently referring to any active object, it should ***always*** be set to "null" – the address zero (0).

```
int *i = 0;
Person *p = 0;
Person *q; // WARNING: q is not
           // pointing to null!
```

Working with Data in C++

- A pointer can be obtained for *any* value – including pointers!
  - This is done with the & operator.

```
Person p("Harrison Ford", 75);
   Person *pPtr = &p;
Person **pPtrPtr = &pPtr;
// Yes, pointers to pointers
// are completely legal.
```

Function Calls

- In C++, each function may specify the manner by which its parameters are received.
  - The type declaration of the parameter determines whether the data is passed "by value" or "by reference."
    - Value types are said to be passed "*call by value*".
    - On the other hand, reference types are said to be passed "*call by reference*."

Call By Value

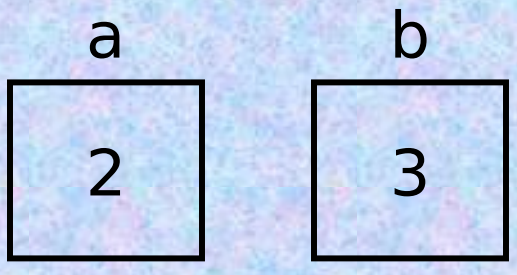
```
void
  swap(int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}
```

```
void main()
{
    int a = 2;
    int b = 3;

    swap(a, b);
}
```

Call By Value

```
void
  swap(int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}
```

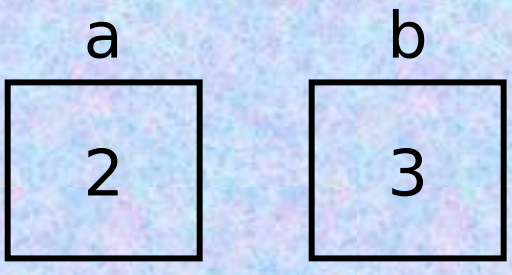
```
void main()
{
    int a = 2;
    int b = 3;

    swap(a, b);
}
```

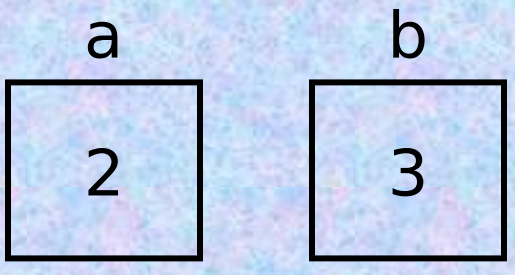


Call By Value

```
void
  swap(int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}
```

```
void main()
{
    int a = 2;
    int b = 3;

    swap(a, b);
}
```

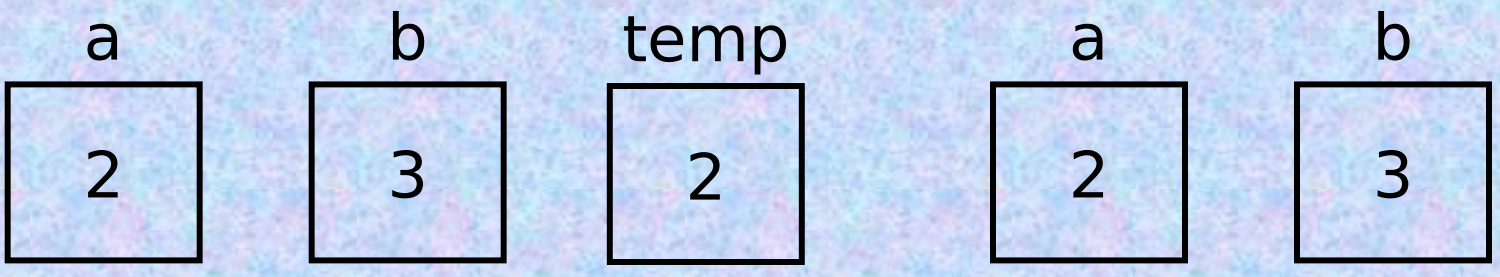




Call By Value

```
void
  swap(int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}
```

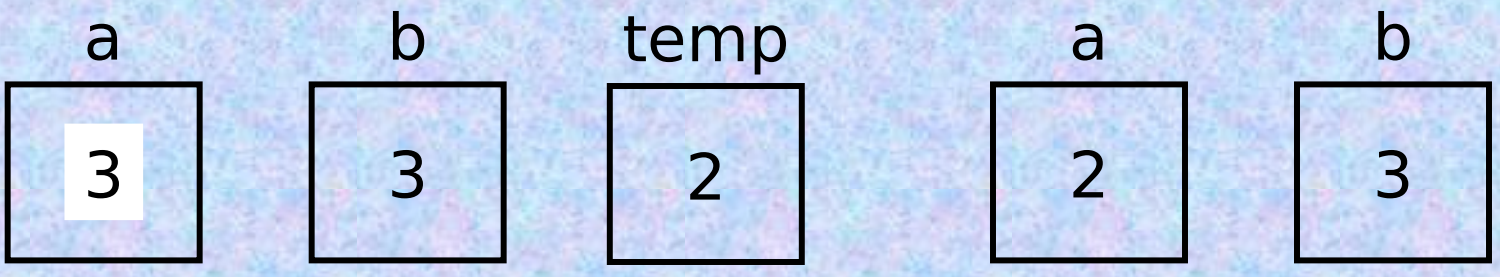
```
void main()
{
    int a = 2;
    int b = 3;

    swap(a, b);
}
```



Call By Value

```
void
  swap(int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}
```

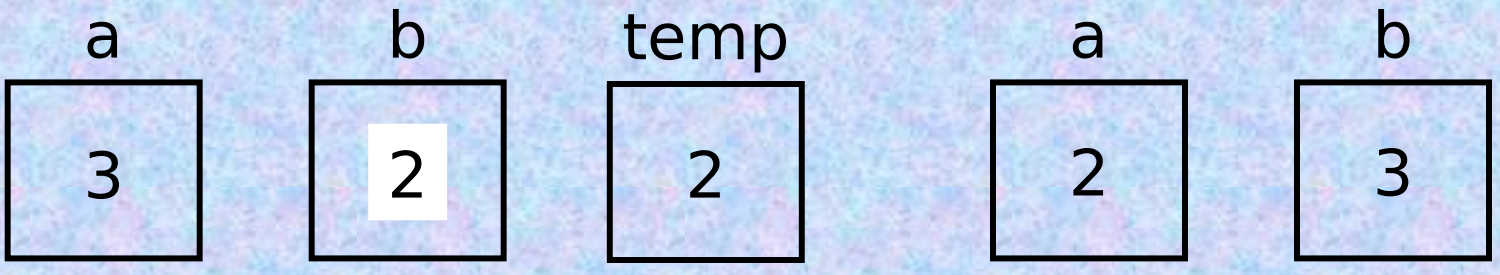
```
void main()
{
    int a = 2;
    int b = 3;

    swap(a, b);
}
```



Call By Value

```
void
  swap(int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}
```

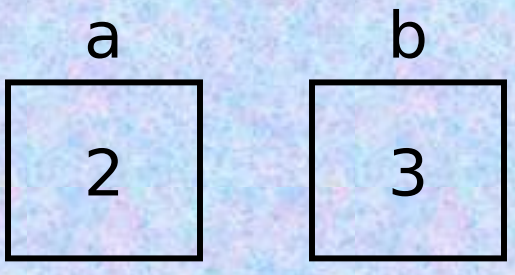
```
void main()
{
    int a = 2;
    int b = 3;

    swap(a, b);
}
```



Call By Value

```
void
  swap(int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}
```

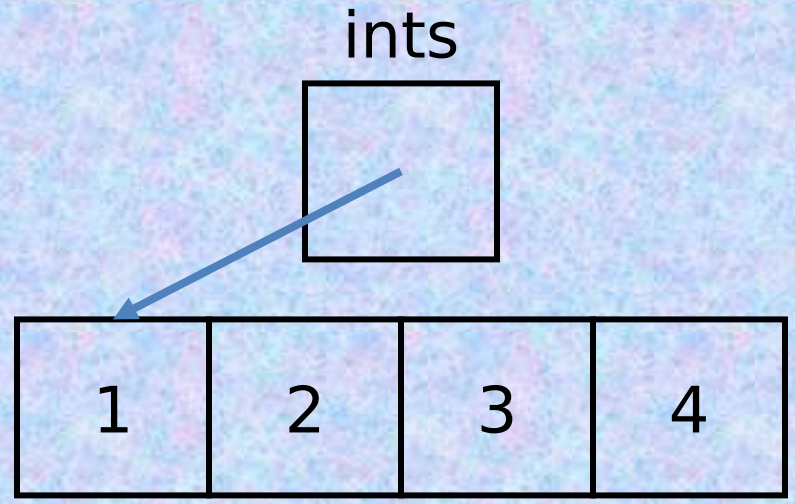
```
void main()
{
    int a = 2;
    int b = 3;

    swap(a, b);
}
```



Call By Reference

```
void swap(int* ints,
 int i_1, int i_2)
{
    int temp =
      ints[i_1];
    ints[i_1] =
      ints[i_2];
    ints[i_2] = temp;
}
```

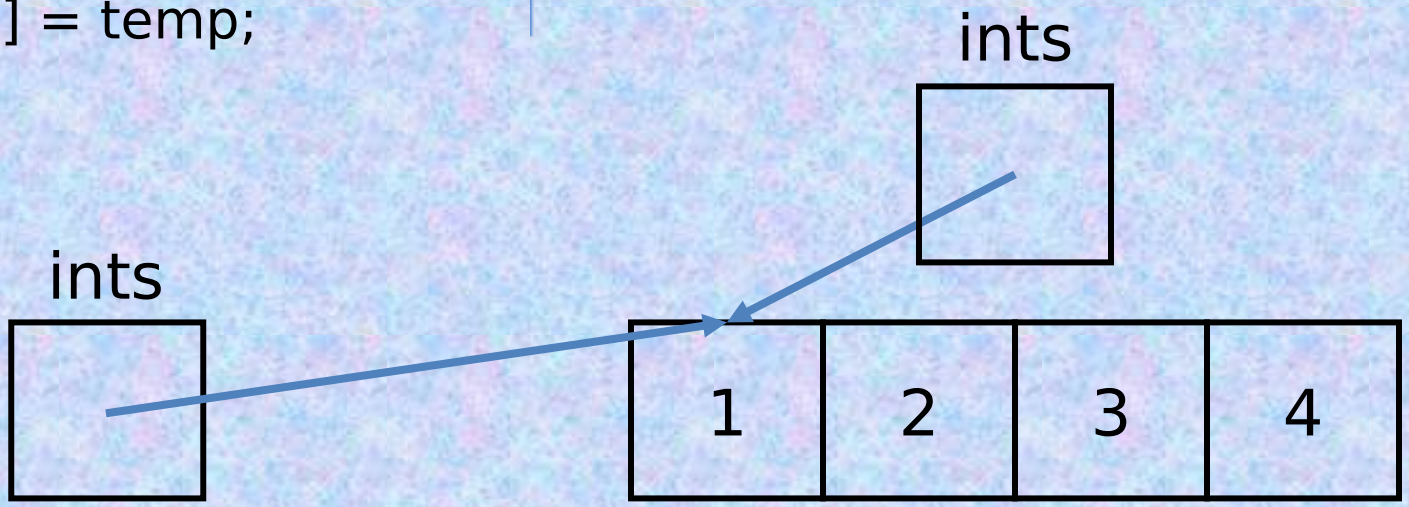
```
public void main()
{
    int[] ints =
      {1, 2, 3, 4};

    swap(ints, 0, 3);
}
```



Call By Reference

```
void swap(int* ints,
 int i_1, int i_2)
{
    int temp =
      ints[i_1];
    ints[i_1] =
      ints[i_2];
    ints[i_2] = temp;
}
```

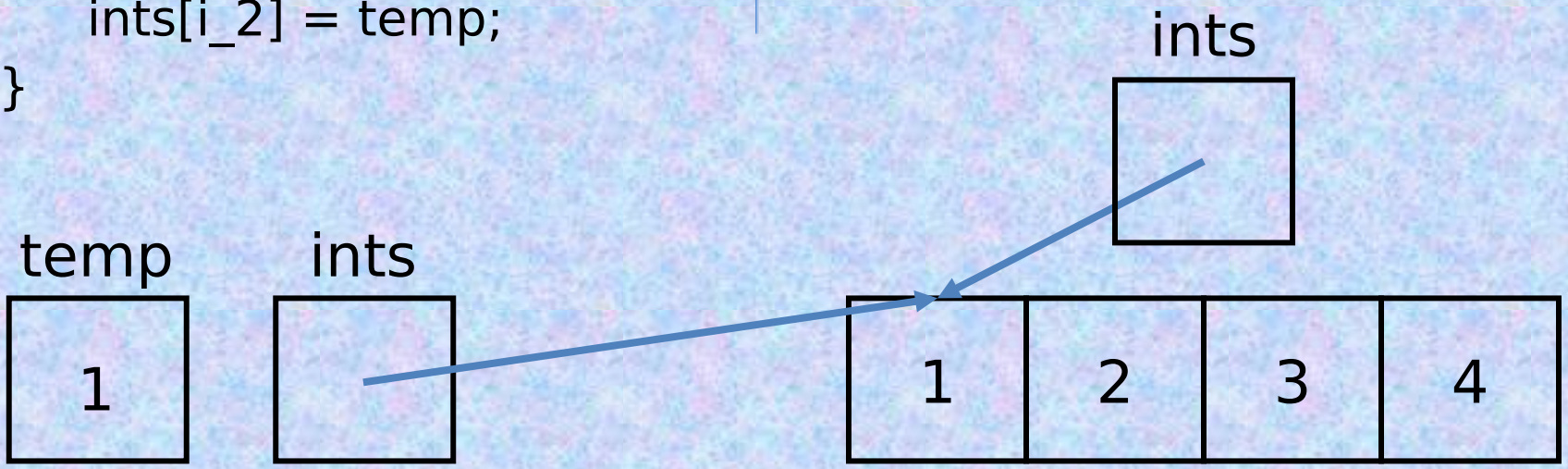
```
public void main()
{
    int[] ints =
      {1, 2, 3, 4};

    swap(ints, 0, 3);
}
```



Call By Reference

```
void swap(int* ints,
 int i_1, int i_2)
{
    int temp =
      ints[i_1];
    ints[i_1] =
      ints[i_2];
    ints[i_2] = temp;
}
```

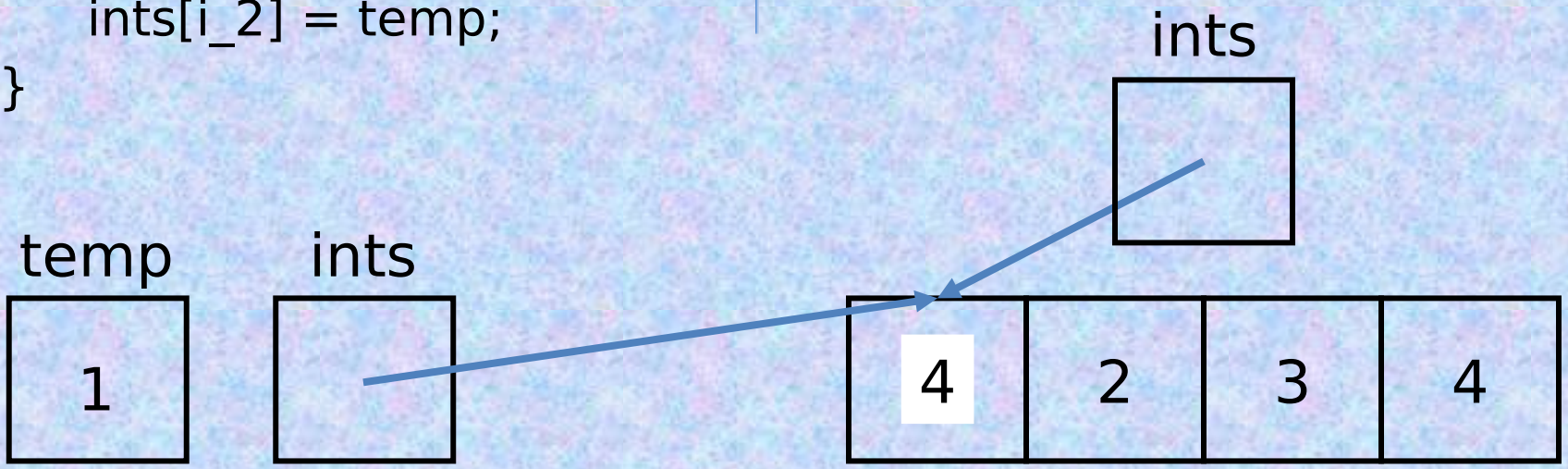
```
public void main()
{
    int[] ints =
      {1, 2, 3, 4};

    swap(ints, 0, 3);
}
```



Call By Reference

```
void swap(int* ints,
 int i_1, int i_2)
{
     int temp =
       ints[i_1];
     ints[i_1] =
       ints[i_2];
     ints[i_2] = temp;
}
```

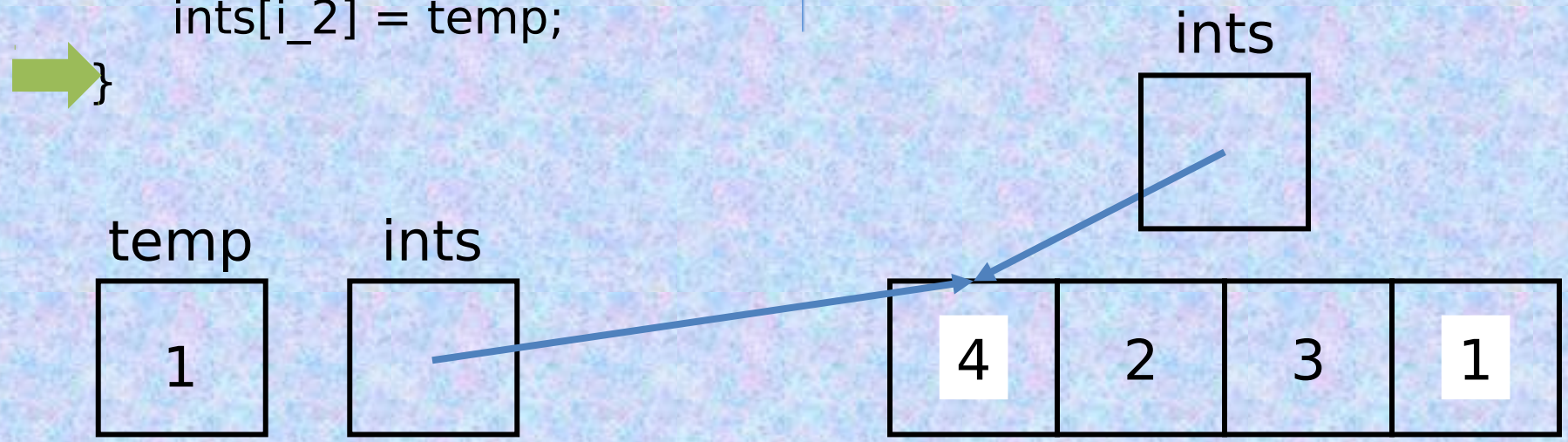
```
public void main()
{
     int[] ints =
       {1, 2, 3, 4};

     swap(ints, 0, 3);
}
```



Call By Reference

```
void swap(int* ints,
 int i_1, int i_2)
{
    int temp =
      ints[i_1];
    ints[i_1] =
      ints[i_2];
    ints[i_2] = temp;
}
```

```
public void main()
{
    int[] ints =
      {1, 2, 3, 4};

    swap(ints, 0, 3);
}
```



Call By Reference

```
void swap(int* ints,
 int i_1, int i_2)
{
     int temp =
       ints[i_1];
     ints[i_1] =
       ints[i_2];
     ints[i_2] = temp;
}
```

```
public void main()
{
     int[] ints =
       {1, 2, 3, 4};

     swap(ints, 0, 3);
}
```