The "const" Keyword with Pointers

**Pointer to Constant**

```c
const int i1 = 100;
cost int i2 = 200;
const int *pi = &i1;          /* pi is a "pointer to a constant." */
i1 = 300;                     /* Error. Cannot change i1. */
i2 = 400;                     /* Error. Cannot change i2. */
pi = &i2;                     /* OK. */
*pi = 500;                    /* Error. Cannot change *pi. */
```

**Constant Pointer**

```c
int i1 = 100;
int i2 = 200;
int *const pi = &i1;          /* pi is a "constant pointer." */
i1 = 300;                     /* OK. */
i2 = 400;                     /* OK. */
pi = &i2;                     /* Error. Cannot change pi. */
*pi = 500;                    /* OK. */
```

**Constant Pointer to Constant**

```c
const int i1 = 100;
cost int i2 = 200;
const int *const pi = &i1;    /* pi is a "constant pointer to a constant." */
i1 = 300;                     /* Error. Cannot change i1. */
i2 = 400;                     /* Error. Cannot change i2. */
pi = &i2;                     /* Error. Cannot change pi. */
*pi = 500;                    /* Error. Cannot change *pi. */
```
Disallowed Mismatch

const int i1 = 100;
const int i2 = 200;
int *pi = &i1;    /* Error. Subversive. Subsequently changing *pi would change i1. */

Disallowed Mismatch in Function Calls

void f(int *pi)
{
    ...
}

...
const int i1 = 5;
const int *pi2 = &i1;
f(pi2);      /* Error. Subversive. If f() changes *pi, then *pi2 also would change. */

Allowed Mismatch

int i1 = 100;
int i2 = 200;
const int *pi = &i1;    /* OK, even though subsequently changing i1 would change *pi. */
i1 = 300;    /* OK. Also changes *pi. */
i2 = 400;    /* OK. */
pi = &i2;    /* OK, even though subsequently changing i2 would change *pi. */
*pi = 500;    /* Error. Cannot change *pi. */

Allowed Mismatch in Function Calls

void f(const int *pi)
{
    ...
}

...
int i1 = 5;
int *pi2 = &i1;
f(pi2);      /* OK. *pi2 is protected against accidental change by f(). */