Module 23: Pass by Value and Pass by Reference

Intro to Computer Science 1 - C++

Professor Scott Frees

Textbook

This topic is covered in section 6.12 of the text

Variable Scope

- Every variable has a scope.
  - A local variable is defined within a segment of code, such as a function, loop, or if block.
  - Local Variables are only visible within the segment they are defined (and sub-segments)

- A global variable is defined outside all functions - and is visible to all functions.
  - You should never use global variables.
  - Exception: Constants can be defined globally

```
const double PI = 3.14159;
```

Variable Scope

Parameters are local variables too

```
void doThis(int x);
```

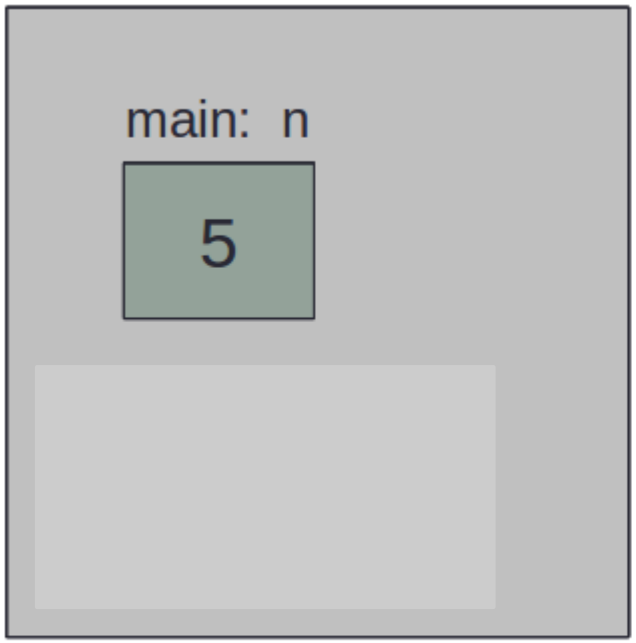
The difference is that their values are initialized when called

```
doThis(5);
```

Never forget that they are local variables though…

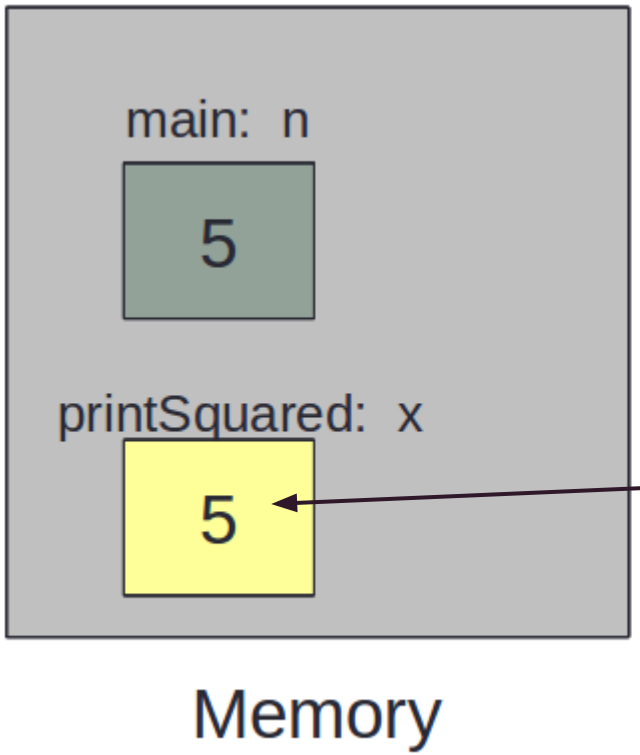
1. They are destroyed when the function returns… along with any changes you made to them.
2. This is called "pass by value" or "pass by copy"

Pass by value



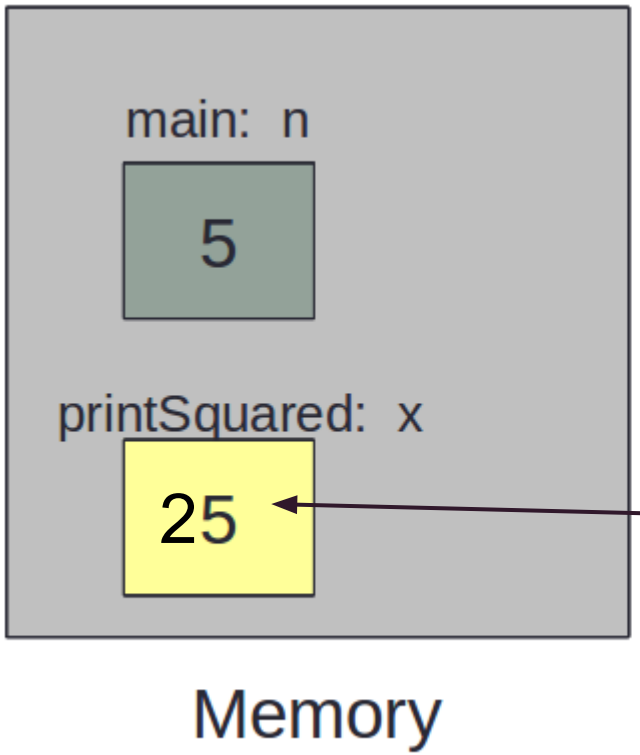

```
int main() {
 int n = 5;
 printSquared(n);
 cout << n << endl;
}
void printSquared(int x) {
 x *= x;
 cout << x << endl;
}
```

Pass by value



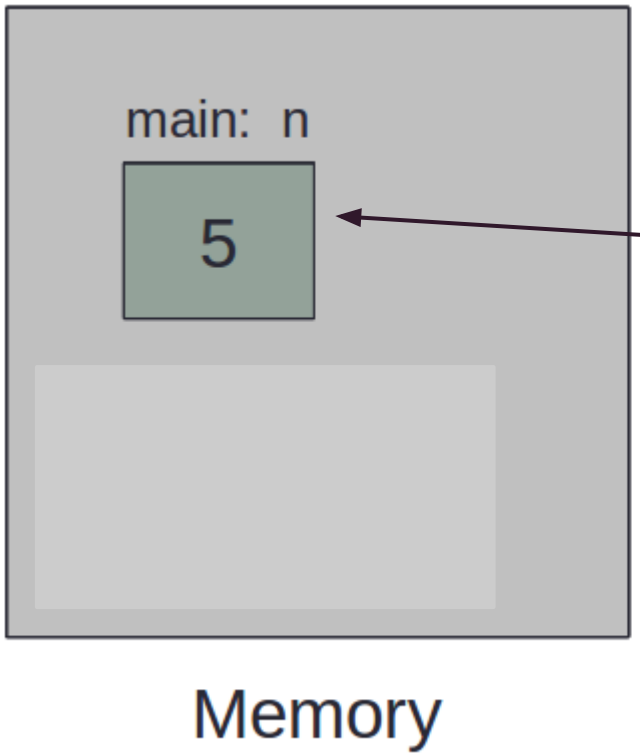

```
int main() {
  int n = 5;
  printSquared(n);
  cout << n << endl;
}
void printSquared(int x) {
  x *= x;
  cout << x << endl;
}
```

Pass by value



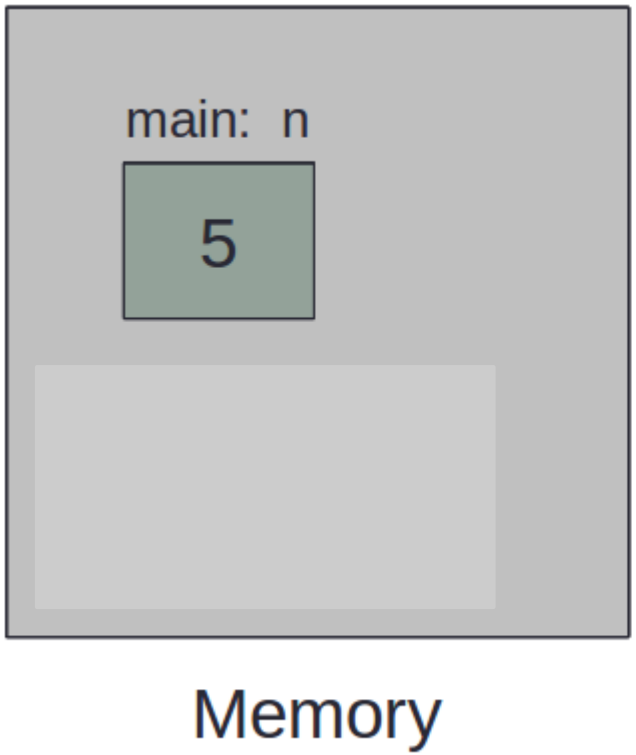

```
int main() {
  int n = 5;
  printSquared(n);
  cout << n << endl;
}
void printSquared(int x) {
  x *= x;
  cout << x << endl;
}
```

Pass by value



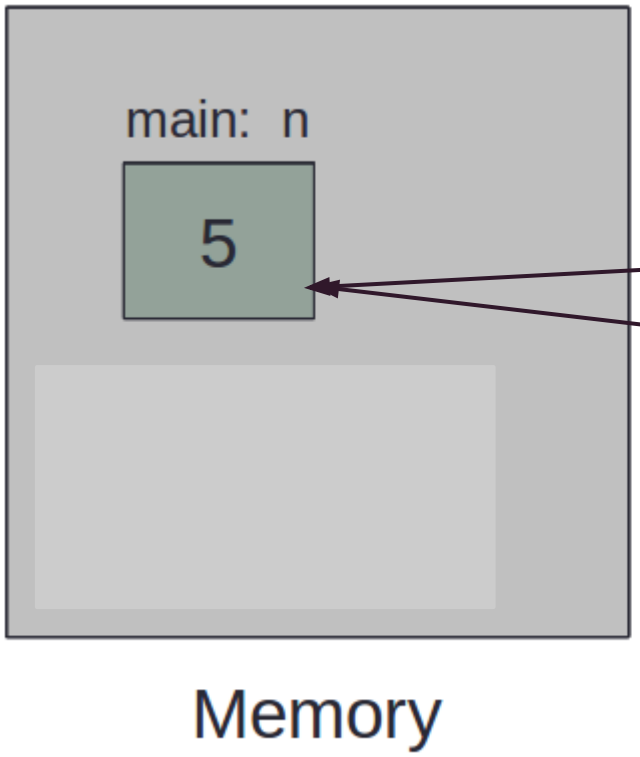

```
int main() {
  int n = 5;
  printSquared(n);
  cout << n << endl;
}
void printSquared(int x) {
  x *= x;
  cout << x << endl;
}
```

Pass by reference



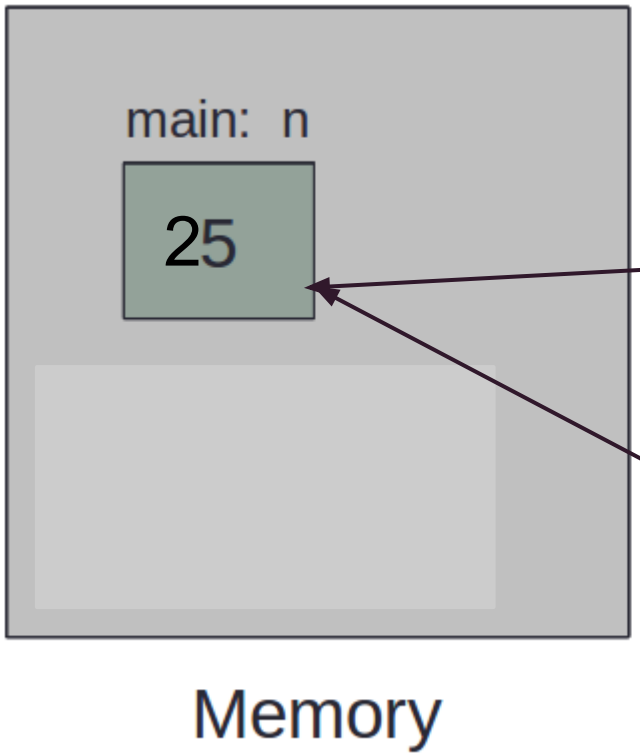

```
int main() {
 int n = 5;
 printSquared(n);
 cout << n << endl;
}
void printSquared(int&x) {
 x *= x;
 cout << x << endl;
}
```

Pass by reference



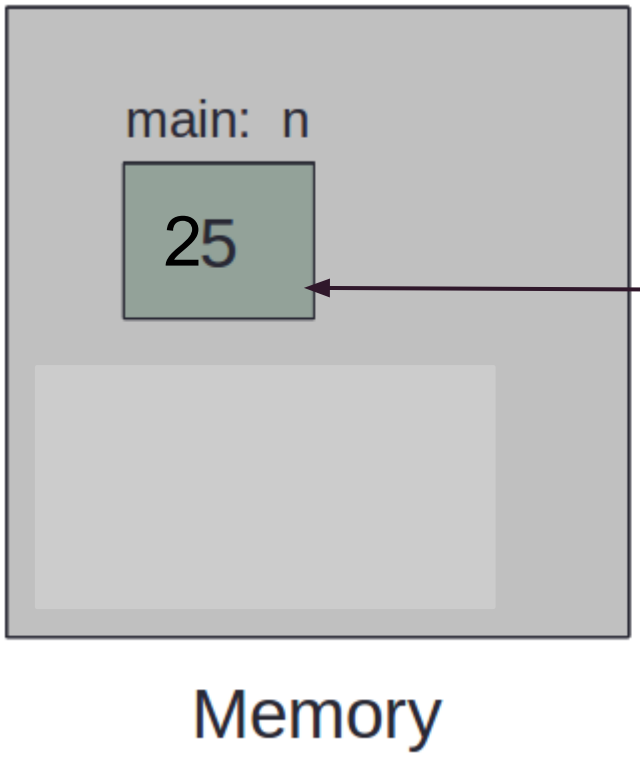

```
int main() {
  int n = 5;
  printSquared(n);
  cout << n << endl;
}
void printSquared(int&x) {
  x *= x;
  cout << x << endl;
}
```

Pass by reference




```
int main() {
  int n = 5;
  printSquared(n);
  cout << n << endl;
}
void printSquared(int&x) {
  x *= x;
  cout << x << endl;
}
```

Pass by reference




```
int main() {
  int n = 5;
  printSquared(n);
  cout << n << endl;
}
void printSquared(int&x) {
  x *= x;
  cout << x << endl;
}
```

When to use reference?

Pass by reference should be used only when necessary!

- Its confusing, and can lead to bugs if not done properly

Its necessary when:

1. You need an input variable to change within a function, and have that change available after the function is called.
2. You need to return two “things” from a function
3. You want to avoid overhead of a copy (later)

Programming Example 27

Ask use for number between 1 and 99

calculate the most efficient coin usage:

86 cents = 3 quarters, 1 dime, 1 penny

```
int computCoin (int coinValue, int & amountLeft)
```

returns number of coins

ie. 25, 10, 5, etc

amount of change left will be updated