Pointers I

- Say we have a dog class:
  - get_age() – gets the age of a dog
  - adopt() – adopts the dog (all dogs start out as not adopted)
  - is_adopted() return true/false depending on if the dog is adopted or not.
  - print()
- I want to write a program that adopts all the puppies in a shelter.

```
vector<dog> alldogs;


for (int x = 0; x < alldogs.size(); x++) {
  if (alldogs[x].get_age() < 1)
    alldogs[x].adopt();
}
```

```
for (int x = 0; x < alldogs.size(); x++) {
  if (alldogs[x].is_adopted())
    alldogs[x].print()
}
```

```
vector<dog> alldogs;
vector<dog> puppies;

for (int x = 0; x < alldogs.size(); x++) {
  if (alldogs[x].get_age() < 1)
    puppies.push_back(alldogs[x])
}

for (int x = 0; x < puppies.size(); x++) {
  puppies[x].adopt();
}

for (int x = 0; x < alldogs.size(); x++) {
  if (alldogs[x].is_adopted())
    alldogs[x].print()
}
```

```
int main()
{
   int x;
   f();
   return 0;
}

void f()
{
   long x;
   double y;
}
```

- Create a new program.
- Declare two integer variables, x and y, set to 5 and 10.
- Print the values of x and y.
- Print the addresses of x and y.
- Make two separate pointers, a and b.
  - Point a to x and b to y.
- Using a and b, print the values of x and y.
- Make the pointers point to the opposite integers.
  - (Make a point to y and b point to x.)
- Print the values of x and y again, using a and b.