Selection sort is one of the simplest algorithms for sorting arrays or vectors. It is rather inefficient, but it is effective for small data sets.

The following code implements a selection sort on an STL vector of PostNet objects, assuming that each PostNet object can return an integer value based on the zip code.

```
// selection sort based on zip code order
void sort(vector<PostNet>& c)
{
	for (unsigned short i=0; i<c.size(); i++)
	{
		// look for smallest zip code in remaining elements
		long min = c[i].getNumericZip();
		unsigned short min_i = i;
		for (unsigned short j=i+1; j<c.size(); j++)
		{
			if (c[j].getNumericZip() < min)
			{
				min = c[j].getNumericZip();
				min_i = j;
			}
		}
		// move the object with the smallest zip code
		// to the front of the remaining elements
		if (min_i != i)
		{
			PostNet tmp = c[i];
			c[i] = c[min_i];
			c[min_i] = tmp;
		}
	}
}
```

If you are using an array of objects, rather than an STL vector, add a size parameter and correct the stopping conditions for both loops.

Assuming that you are storing zip codes as character strings, the following method will convert zip codes to equivalent integers:

```
long PostNet::getNumericZip()
{
	return	(zipcode[0]-'0')*10000
		+ (zipcode[1]-'0')*1000
		+ (zipcode[2]-'0')*100
		+ (zipcode[3]-'0')*10
		+ (zipcode[4]-'0');
}
```