Variable definitions inside repeat statements

Abhiram Ranade

Program to add numbers

```
main_program{
  float sum=0, term;
  int count; cin >> count;
  repeat(count){
     cin >> term;
     sum += term;
  }
  cout << sum << endl;
}
```

How professionals might write

```
main_program{
  float sum=0;
  int count; cin >> count;
  repeat(count){
    float term;  cin >> term;
    sum += term;
  }
  cout << sum << endl;
}
```

Interpretation

- Variable is created when control reaches the definition.
- Variable is destroyed when control reaches the end of the repeat body.
- term will be created and destroyed in each iteration.
- Program will work fine.
- Motivation: Move definition close to use. Easier to read the program.

General Principle 1

- Block: region from { to }

  examples: main_program, body of repeat
- Parent block of definition: innermost block containing definition.
  - Parent block of float sum=0; : main_program
  - Parent block of float term; : body of repeat
- Variable created when control reaches definition, destroyed when control reaches end of parent block.
- Applicable to repeats inside repeats, other blocks that we will study later.

Multiple definitions of same name

- Same name may be defined several times in the same program, provided all definitions have different parent blocks.
- Non-example:

```
main_program{
  int sum;
  repeat{  ....
  }
  int sum;  // not allowed.
}
```

Example 1

```
main_program{
  repeat(3){
    int p=5;  cout << p << endl;
  }
  repeat(2){
    int p=6;  cout << p << endl;
  }
}
```

Interpretation

- Variable p created when first loop executes, initialized to 5.
- Variable p created when second loop executes, initialized to 6.
- Will print 5 5 5 6 6
- Analogy:
  - Two families have sons with name “Raju”
  - If you mention Raju in one of the families, you will be referring to the son in that family.

Analogy 2

- Some family has son named Manmohan.
- Within that family, “Manmohan” refers to the son in that family.
- In other families, “Manmohan” refers to the Prime Minister.
- Prime minister overshadowed.
- Something similar in programming also.

Example 2

```
main_program{
  int M=86;
  repeat(3){
      int M = 2; // Will shadow previous defn.
      cout << M << endl;
  }
  cout << M << endl;
}  // will print out 2  2  2  86
```

Example 3

```
main_program{
  int M=86;
  repeat(3){
      cout << M << endl;
      int M = 2;
      cout << M << endl;
  }
  cout << M << endl;
}  // will print out 86 2  86 2  86 2  86
```

Summary

- Move definitions close to their use.