INF3110

Exercises week 28.9 - 2.10, 2009

Problem 1

Exercise 7.1 in Mitchell.

Problem 2

Exercise 7.8 in Mitchell.

Problem 3

```
{                      --block 1
  int i, j, k;         --1
  ...
  {                    --block 2
    int i, k;          --2
    ...
    {                  --block 3
      int j;           --3
      ...
    }                  --end block 3
    ...
    {                  --block 4
      int i, l;        --4
      ...
    }                  --end block 4
  }                    --end block 2
  ...
  {                    --block 5
   int a, b, c, d;     --5
   ...
  }                    --end block 5
}                      --end block 1
```

Describe scope and life-time of the various variables wrt to the different blocks. Use the names of the blocks and the numbers of interesting lines.

Problem 4

```
{
  int i=1, j=2, k=3;
  alpha() {
   int i=4, l=5;
   i=k+l;
                     -- **
   beta();
  };
  beta() {
    int k=6:
    i=j+k;
                     -- *
    alpha();
  };
  main() {
    beta ();
  }
}
```

Execution starts by executing main().

Draw the run-time stack at three points of execution:
 1) first time execution reaches the line marked with *,
 2) when execution reaches the line marked with **, and
 3) second time execution reaches the line marked with *

Show access and control links, and values of variables. Assume that the language is statically scoped.

As execution starts with execution of main, the control link of the main activation record is of no significance.

Problem 5

```
{
  f2() {
    int i;
    ...
    ... i ...
    ... j ...
    ...
  }
  f1() {
    int j;
    ...
    ... i ...
    ... k ...
    f2();
    ...
  }
  main() {
    int i, j, k;
    k = 0;
    i = 5;
    j = 7;
    f1();
    f2();
    ...
  }
}
```

Assume that this is written in a language with dynamic scoping. What will happen at the two calls f1() and f2(): to which declarations will the applications of i, j, k within f1 and f2 be bound?

Problem 6
This is an example in a language with static scoping.

```
{
  int x, y, z;
  f1(){
    int t, u;
    f2(){
      int x, w;
      f3(){
        int y, w, t;
        ... ;
        f2();
        ...
      };
      x = y + t + w + z;
      f3();
    }
    ...;
    f2();
    ...
  }
  main(){
    int z, t;
    ...
    f1();
    ...
  }
}
```

Show the run-time stack with both control and access links for the following call sequence: main; f1(); f2(); f3(); f2().

Explain what happens with variable bindings when executing ‘x = y + t + w + z’ in the latest call of f2. Look especially at y, t and z.

Problem 7

Exercise 7.4 a) in Mitchell.

Problem 8

Exercise 7.7 in Mitchell.

Problem 9

Consider the example below. Discuss call by reference and call by value-result for swap(a[i], a[j]). What happens if i=j?

```
swap(int x, int y) {
  x = x + y;
  y = x – y;
  x = x – y;
}
```