

Concepts of programming languages essay sample

[Technology](#), [Computer](#)



1. Compute the weakest precondition for each of the following assignment statements and postconditions: (p. 165 Problem set 19, 20)

$$(a) a = 2 * (b - 1) - 1 \{a > 0\}$$

$$(b) a = a + 2 * b - 1 \{a > 1\}$$

$$(c) a = 2 * b + 1;$$

$$b = a - 3$$

$$\{b < 0\}$$

$$(d) a = 3 * (2 * b + a);$$

$$b = 2 * a - 1$$

$$\{b > 5\}$$

$$(a) 2 * (b - 1) - 1 > 0$$

$$2 * (b - 1) > 1$$

$$2b - 2 > 1$$

$$2b > 3$$

$$\{b > 1.5\}$$

$$(b) a + 2 * b - 1 > 1$$

$$a + 2 * b > 2$$

$$\{a > 2 - 2 * b\}$$

(c) $a - 3 < 0$

a^3

$2 * b + a > 1$

$\{2 * b > 1 - a\}$

2. Write a denotational semantics mapping function for the following statements: (p. 165 Problem set 21)

(a) Java for

(b) C do-while

(a)

= for (, ,) { }

=

=

=

= $> | < | \geq | 0$

count = n;

sum = 0;

while count 0 do

sum = sum + count;

count = count - 1;

end

$$\{sum = 1 + 2 + \dots + n\}$$

part of the invariant can be:

$$sum = (count+1) + (count+2) + \dots + (n-1) + n, \text{ let's call this } I$$

Thus we have I and $\{n > 0\}$, and $\{count > 0\}$, which reduce to I and $\{count > 0\}$

Because in Loop we have $(count = count - 1)$, and $\{count > 0\}$, thus the precondition before the loop is $\{count \geq 1\}$

Using this precondition for the first assignment in Loop, we have $sum = sum + count$ $\{sum = count + (count + 1) + \dots + n \text{ and } count \geq 1\}$

The precondition for this is $sum = 1 + 2 + \dots + n$. thus we proved that the post-condition for this program is correct.

4. Dynamic type binding is closely related to implicit heap-dynamic variables. Explain this relationship. (p. 241 Problem set 5)

In languages featuring dynamic binding, most of them supports virtual memory management, in which the virtual machine determines the variable type, allocated memory space and de-allocate them after the variable is no longer needed. These are implicit heap-dynamic variables where users do not have to worry about what type of variables they are going to use, and save a lot of programmer time for de-allocating un-used memory spaces.

5. Consider the following skeletal C program: (p. 244 Problem set 13)

```
void fun1(void);/* prototype */
```

```
void fun2(void);/* prototype */
```

```
void fun3(void);/* prototype */
```

```
void main() {
```

```
int a, b, c;
```

```
...
```

```
}
```

```
void fun1() {
```

```
int b, c, d;
```

```
...
```

```
}
```

```
void fun2() {
```

```
int c, d, e;
```

```
...
```

```
}
```

```
void fun3() {
```

```
int d, e, f;
```

```
...
```

```
}
```

Given the following calling sequences and assuming that dynamic scoping is used, what variables are visible during execution of the last function called? (Also state the names of the functions in which the visible variables are defined.)

(a) main calls fun1; fun1 calls fun2; fun2 calls fun3.

main(a), f1(b), f2(c), f3(d, e, f)

(b) main calls fun3; fun3 calls fun1.

f1(b, c, d), f3(e, f), main(a)

6. Give the representations of the following numbers as 16-bit binary numbers, twos complement for negative numbers:

1. 248 - 0000000011111000

2. -248 1111111100001000

3. -8412 1101111100100100

7. Convert the following float numbers expressed by binary into decimal expression, using IEEE single precision floating point standard representation.

1. 01111111100000000000000000000000 sign bit 0, e is 255, f is 0, therefore this is +infinity
2. 11111111100100010001001010101010 e is 255 and f is non zero, this is no a number
3.

00000000100000000000000000000000 this is 1. 1754943508222875e-38

8. Multidimensional arrays can be stored in row major order, as in Pascal, or in column major order, as in FORTRAN. Develop the access functions for both of these arrangements for three-dimensional arrays. (p. 307 Problem set 10)

let's assume the index starts from 1, and size of x by y by z

row major:

```
int Access(i, j, k)
{
return (k-1)*y*x+(j-1)*x+i;
}
```

column major:

```
int Access(i, j, k)
{
return (k-1)*y*x+(i-1)*x +j;
}
```