

Code Excerpts referred to in Practice Exam Questions

p01.cc

```
1 // p01.cc    CISC181 Final Exam Fall 2005
2
3 #include <iostream>
4 using namespace std;
5
6 void funky(int x);
7
8 int main(void)
9 {
10     int a = 2;
11     funky(a);
12     cout << "a=" << a << endl;
13     return 0;
14 }
15
16 void funky(int x)
17 {
18     x = x * 3;
19     cout << "x=" << x << endl;
20 }
21
22
23
24
```

p02.cc

```
1 // p02.cc   CISC181 Final Exam Fall 2005
2
3 #include <iostream>
4 using namespace std;
5
6 void fresh(int &y);
7
8 int main(void)
9 {
10     int b = 3;
11     fresh(b);
12     cout << "b=" << b << endl;
13     return 0;
14 }
15
16 void fresh(int &y)
17 {
18     y = y * 5;
19     cout << "y=" << y << endl;
20 }
21
22
23
24
```

CISC 181 Midterm Practice-Parameter Passing

Questions 1 through 5 deal with code in q5 . cpp, shown here:

```
// q5.cpp for CISC181 exam E01, Fall 2005
// P. Conrad, 10/07/2005

#include <iostream>
using std::cout;
using std::endl;

void incl(int n, int m)
{
    n++;
    m++;
    cout << "incl: " << m << endl;
}

int main()
{
    int a = 3;
    int b = 4;
    int c = 5;

    cout << "main1: " << a << " " << b << endl;
    incl(a,b);
    cout << "main2: " << a << " " << b << endl;

    return 0;
}
```

1. (3 pts) Which of the following are the *names* of the *formal parameters* of function `incl()` ?
 - (a) a and b
 - (b) n and m
 - (c) int and int
 - (d) `std::cout` and `std::endl`
 - (e) `n++` and `m++`
2. (3 pts) Which of the following are the *names* of the *actual parameters* of function `incl()` ?
 - (a) a and b
 - (b) n and m
 - (c) int and int
 - (d) `std::cout` and `std::endl`
 - (e) `n++` and `m++`

3. (4 pts) Which of the following would be the *first* line of output from this program?
- (a) `incl: 4`
 - (b) `incl: 5`
 - (c) `main1: 3 4`
 - (d) `main2: 3 4`
 - (e) `main2: 4 5`
4. (4 pts) Which of the following would be the *last* line of output from this program?
- (a) `incl: 4`
 - (b) `incl: 5`
 - (c) `main1: 3 4`
 - (d) `main2: 3 4`
 - (e) `main2: 4 5`
5. (3 pts) Which of the following are the *types* of the *actual parameters* of function `incl()` ?
- (a) `a` and `b`
 - (b) `m` and `n`
 - (c) `int` and `int`
 - (d) `std::cout` and `std::endl`
 - (e) `n++` and `m++`

Here is a sample program, followed by some sample output.—except, the numbers have been replaced with blanks First, trace through the program to determine what goes in each of these blanks, then select the correct answers from the choices given in the problems.

Don't panic: Each of the sections of code can be done independently of the other sections, so if you are stuck on one, don't stop; keep working.

```
1 /* ql0.cc Agnes Nitt for CISC181 final exam */
2
3 #include <iostream>
4 using namespace std;
5
6 void trouble(int *r);
7 double fire(double f);
8 int burn(int b, int *n);
9 void cauldron(int &c, int &d);
10
11 int main(void)
12 {
13     int e=2, f=3, t = 4, y = 9;
14     double v=5.1, z = 10.2;
15
16     trouble(&y);
17     cout << "y=" << y << endl;
18
19     z = fire(v);
20     cout << "v=" << v << " z=" << z << endl;
21
22     t = 3;
23     burn(5, &t);
24     cout << "t=" << t << endl;
25
26     e=4;
27     f=5;
28     cauldron(e,f);
29     cout << "e=" << e << " f=" << f << endl;
30     return 0;
31 }
32
33 void trouble(int *r)
34 {
35     (*r) = -1;
36     return;
37 }
38
39 void cauldron(int &c, int &d)
40 {
41     int temp = c;
42     c = d;
43     d = temp;
44 }
45
46 int burn(int b, int *n)
47 {
48     (*n) = b;
49 }
50
51 double fire(double f)
52 {
53     return (f + 0.1);
54 }
```

```
> ./q10
y=___
v=___ z=___
t=___
e=___ f=___
>
```

6. (2 pts) What goes in the blank $y=$ ___ ?

- (a) $y=-1$
- (b) $y=-9$
- (c) $y=9$
- (d) $y=-8$
- (e) none of the above

7. (2 pts) What goes in the blank $v=$ ___ $z=$ ___ ?

- (a) $v=5.1$ $z=5.1$
- (b) $v=5.2$ $z=5.2$
- (c) $v=5.2$ $z=5.1$
- (d) $v=5.1$ $z=5.2$
- (e) none of the above

8. (2 pts) What goes in the blank $t=$ ___ ?

- (a) $t=3$
- (b) $t=5$
- (c) $t=8$
- (d) $t=15$
- (e) none of the above

9. (2 pts) What goes in the blank $e=$ ___ $f=$ ___ ?

- (a) $e=5$ $f=4$
- (b) $e=4$ $f=5$
- (c) $e=5$ $f=5$
- (d) $e=4$ $f=4$
- (e) none of the above

Questions 10 through 15 deal with the code for `p01.c` on page 1.

10. (1 pts) Line 6 of the program `p01.c` contains which of the following?

- (a) a function prototype
- (b) a function definition
- (c) a function call
- (d) a pre-processor directive

11. (1 pts) Line 11 of the program `p01.c` contains which of the following?
- (a) a function prototype
 - (b) a function definition
 - (c) a function call
 - (d) a pre-processor directive
12. (2 pts) The variable `x` in `p01.c` is
- (a) a formal parameter
 - (b) an actual parameter
13. (2 pts) The variable `a` in `p01.c` is
- (a) passed by value
 - (b) passed by reference
14. (2 pts) The output of `p01.c` will be
- (a) `x=6`
`a=6`
 - (b) `a=2`
`x=6`
 - (c) `a=6`
`x=6`
 - (d) `x=6`
`a=2`
15. (2 pts) Suppose line 11 of `p01.c` were replaced with:

```
funky(a+2);
```

What would be the result if you tried to compile and run?

- (a) `x=12`
`a=2`
- (b) `a=2`
`x=12`
- (c) `a=4`
`x=12`
- (d) `x=12`
`a=4`
- (e) You'd get a syntax error, because you can't pass `(a+2)` to `funky`.

Questions 16 through 22 deal with the code for `p02.c` on page 2.

16. (1 pts) Lines 16–20 of the program `p02.c` contain which of the following?
- (a) a function prototype
 - (b) a function definition
 - (c) a function call
 - (d) a pre-processor directive
17. (1 pts) Line 3 of the program `p02.c` contains which of the following?
- (a) a function prototype
 - (b) a function definition
 - (c) a function call
 - (d) a pre-processor directive
18. (2 pts) The variable `b` in `p02.c` is
- (a) a formal parameter
 - (b) an actual parameter
19. (2 pts) The variable `b` in `p02.c` is
- (a) passed by value
 - (b) passed by reference
20. (2 pts) The symbol `&` on line 16 of `p02.c` indicates which of the following:
- (a) take the *address* of `y`
 - (b) that the *type* of `y` is `int *`
 - (c) that `y` is being passed by *reference* to `fresh`
 - (d) that both `int` *and* `y` are parameters of `fresh`
21. (2 pts) The output of `p02.c` will be
- (a) `y=15`
`b=15`
 - (b) `b=15`
`y=15`
 - (c) `b=3`
`y=15`
 - (d) `y=15`
`b=3`

22. (2 pts) Suppose line 11 of `p02.c` were replaced with:

```
fresh(b+2);
```

What would be the result if you tried to compile and run?

- (a) `y=25`
`b=3`
- (b) `y=25`
`b=5`
- (c) `y=25`
`b=25`
- (d) `b=5`
`y=25`
- (e) You'd get a syntax error, because you can't pass `(b+2)` to `fresh`.

end of exam practice questions

1. (b)
2. (a)
3. (c)
4. (d)
5. (c)
6. (a)
7. (d)
8. (b)
9. (a)
10. (a)
11. (c)
12. (a)
13. (a)
14. (d)
15. (a)
16. (b)
17. (d)
18. (b)
19. (b)
20. (c)
21. (a)
22. (e)

end of key