

# Physics 234: Computational Physics

In-class Midterm Exam

Friday, February 12, 2010

Student's Name: \_\_\_\_\_

## **Fill-in-the-blank and multiple choice questions (20 points)**

Mark your answers on the exam sheet in blue or black ink. Please be clear about your selections. Each question is worth one point unless otherwise indicated.

In code listings where the preamble is not written explicitly, you may assume the following.

```
#include <cassert>

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

You may also assume that small snippets of code are properly embedded in a `main` function.

1. What output does this produce?

```
signed char a = 128;
cout << int(a) << ", ";
--a;
cout << int(a) << endl;
```

- (a) 128,127
- (b) 128,-127
- (C) -128,127
- (d) -128,-129

2. What polynomial is computed, and at points on what interval?

```
double horner(double x, const double* a, const double *b)
{ return a < b ? *a + x*horner(x,a+1,b) : 0.0; }
```

```
const double p[4] = { 1.0, -2.0, 0.25, 0.125 };
```

```
int main()
{
    for (int n = 0; n <= 100; ++n)
    {
        const double xn = n/50.0;
        cout << xn << "\t" << horner(xn,p,p+4) << endl;
    }
    return 0;
}
```

- (A)  $1 - 2x + \frac{1}{4}x^2 + \frac{1}{8}x^3$  on the interval  $[0, 2]$
- (b)  $1 - 2x + \frac{1}{4}x^2 + \frac{1}{8}x^3$  on the interval  $[0, 2)$
- (c)  $x^3 - 2x^2 + \frac{1}{4}x + \frac{1}{8}$  on the interval  $[0, 2]$
- (d)  $x^3 - 2x^2 + \frac{1}{4}x + \frac{1}{8}$  on the interval  $[0, 2)$

3. Which one of the following is not a synonym for the same fundamental type. Circle your choice.

short, short int, int, signed short, signed short int

4. What is the value of N just before the function `count_down` returns?

```
void count_down(int &N) { while (N) N--; }
```

```
int main()
{
    count_down(10);
    return 0;
}
```

- (a) 0  
(b) 1  
(c) 10  
(D) program won't compile
5. Which of the following is the correct scenario?

```
int x, y, z;
int main()
{
    x = y = z = 0;
    assert( x == y == z );
    return 0;
}
```

- (a) program compiles, runs, and exits cleanly  
(B) program compiles but fails at runtime on the `assert`  
(c) program won't compile

6. (2 points) Make note of the various syntax errors by (i) circling the incorrect punctuation and (ii) writing in the missing semicolons.

```
#include "iostream"  
using std::cout;  
using std::endl;  
  
int main(int argc, char* argv())  
{  
    for (int i = 0, i < argc, ++i)  
        cout >> argv(i) >> endl;  
    cout >> endl;  
    return 0;  
}
```

7. Write this program's output in the box provided.

```
const int N = 4;  
for (int k = 0; k < N; ++k)  
    for (int j = 0; j < k; ++j)  
        for (int i = 0; i < j; ++i)  
            cout << "(" << i << ", " << j << ", " << k << ")";  
cout << endl;
```

terminal output: (0,1,2)(0,1,3)(0,2,3)(1,2,3)

8. Match each of the following declarations with the most appropriate description. Write the corresponding letter in the space provided.

(a) `int* p;` (b) `double x;` (c) `char a;` (d) `char r[5];`

  b   high-precision floating point number

  c   integer type of exactly one byte in length

  a   pointer to a fundamental type

  d   40 contiguous bits of storage

9. What are the final values of `i` and `x` after the following two statements are executed. (Be sure to express your answers using a literal that is consistent with the type of each variable.)

```
int a = 3, b = -2;
const double x = ++a+-b*5+9 / 5+a;
```

`a == 4` `x == 19.0`

10. Suppose that your computer represents `signed char` internally using the two's complement scheme. Carry out the addition `a+b` by hand and fill in the empty boxes.

```
signed char a = 10;
signed char b = -30;
signed char c = a+b;
```

	<i>decimal</i>	<i>two's complement</i>																									
Hint:	9	$(00001001)_2$																									
	-23	$(11101001)_2$	+																								
	-29	$(11100011)_2$																									
			<table border="1" style="border-collapse: collapse; margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	1	0	1	0	1	1	1	0	0	0	1	0	1	1	1	0	1	1	0	0
0	0	0	0	1	0	1	0																				
1	1	1	0	0	0	1	0																				
1	1	1	0	1	1	0	0																				

11. (2 points) Given the C array definitions,

```
char p[] = { 'p', 'h', 'y', 's', '2', '3', '4', '\0' };
char q[] = "phys123";
```

what do each of the following equalities evaluate to? In each case, circle either true (t), false (f), or unpredictable (u).

- (t / f / **U**) `p < q`;  
 (t / **F** / u) `*p < *q`;  
 (**T** / f / u) `*(p+6) != *(q+6)`;  
 (**T** / f / u) `*(p+7) == *(q+7)`;

12. This program computes the matrix product  $C = AB$ . Matrix elements are stored in flat arrays arranged in row-major order. What's the correct body of the `index` function?

```
int index(int i, int j, int M) { ... }

void mult(double out[], const double inL[], const double inR[],
          int N, int K, int M)
{
    for (int i = 0; i < N; ++i)
        for (int j = 0; j < M; ++j)
            {
                double sum = 0.0;
                for (int k = 0; k < K; ++k)
                    sum += inL[index(i,k,K)]*inR[index(k,j,M)];
                out[index(i,j,M)] = sum;
            }
}

int main()
{
    const double A[3*4] = { 1.0, 0.0, 0.0, 0.0,
                           0.0, 1.0, 0.0, 0.0,
                           0.0, 0.0, 1.0, 0.0 };
    const double B[4*2] = { -3.0, 0.1,
                           -4.0, 0.25,
                           1.5, 3.6,
                           2.25, 0.0 };

    double C[3*2];
    mult(C,A,B,3,4,2);
    return 0;
}
```

- (A) `return i*M+j;`  
(b) `return i+j*M;`  
(c) `return (i-1)*M+j-1;`  
(d) `return i*j+M;`

13. Suppose I invoke a program by typing this at the command line:

```
$ ./myprog -[ ]verbose -n [5] file1.dat file2.dat
```

Circle the two characters that are accessible as `argv[1][1]` and `argv[3][0]` from within `main(int argc, char* argv[])`.

14. What is the best mathematical interpretation of the calculation carried out by the function call `bf(-3.0,1.5,6)`?

```
double bf(double E, double kT, long unsigned int n)
{
    const double x = -E/kT;
    E = 1.0;
    while (n > 0) (E *= x/n--) += 1;
    return E;
}
```

- (a)  $2^1 + \frac{2^2}{2!} + \frac{2^3}{3!} + \frac{2^4}{4!} + \frac{2^5}{5!} + \frac{2^6}{6!}$   
(b)  $1 + 2 + \frac{4}{2!} + \frac{8}{3!} + \frac{16}{4!} + \frac{32}{5!}$   
(c)  $1 - 2 + \frac{4}{2!} - \frac{8}{3!} + \frac{16}{4!} - \frac{32}{5!} + \frac{64}{6!}$   
(D)  $1 + 2 + 2 + \frac{4}{3} + \frac{2}{3} + \frac{4}{15} + \frac{4}{45}$

15. This program builds a histogram with 100 bins equally-spaced between a and b. What's the correct definition of the bin index i?

```
unsigned long int hist[100], n = 0;
const double a = 1.0, b = 12.5;
double d;
while(cin >> d) { // read in fp values from stdin
    const int i = ... ;
    if (i < 100) { ++hist[i]; ++n; }
}
```

- (a)  $d/100/(a+b)$ ;  
(b)  $d*100/(b-a)$ ;  
(C)  $(d-a)/(b-a)*100$ ;  
(d)  $d/100$ ;

16. (3 points) Suppose we have defined a C string

```
const char config[] = "+-++++-+-+--+---++++-";
```

that encodes the spin-up or spin-down character of a linear chain of localized electrons. We want to compose a function named `ferro` that counts the number of neighbouring electrons that are similarly directed. The function we come up with is the following.

```
int ferro(const char* const p)
{
    int num = 0;
    while (**p != '\0') num += *(p-1) == *p;
    return num;
}
```

Unfortunately, this first attempt isn't quite right.

- (i) Cross out the one `const` modifier that prevents this program from working correctly.
- (ii) Will the statement `ferro(config);` still execute if we remove *both* `const` modifiers? (yes / **NO**)
- (iii) Will the program work identically if `while(**p != '\0')` is replaced by `while(*++p)`? (**YES** / no)