

Physics 5 MidTerm Test Solutions – Fall 08

1. The following program has some errors. Locate as many as you can. Refer to the line numbers.

```
1.  #include <iostream>;
2.  using namespace std;
3.  Int main()
4.  {
5.      char name, go;
6.      cout << "Enter your name: ";
7.      cin >> setw(20);
8.      cin.getline >> name;
9.      cout << "Hi" << name << endl;
10.     cout << "Press the ENTER key to end this program.";
11.     cin >> go;
12.     return 0;
13. }
```

SOLN:

- (1) The semicolon on line 1 is a syntax error
- (2) Int is capitalized on line 3 – a syntax error.
- (3) The “cin >> setw(20);” on line 7 is a syntax error which should just be deleted.
- (4) “cin.getline >> name;” on line 8 is a syntax error, it should read something like
cin.getline(name,20); but this also means that
- (5) the declaration of name on line 5 should be “char name[20];”
- (6) There is a style error on line 9 since the output will look funny if there is no space after “Hi”.
- (7) Line 11 should read “cin.get(go)” if it is to end when the user presses return.

The corrected code might look like this:

```
#include <iostream>
using namespace std;
int main()
{
    char name[20], go;
    cout << "Enter your name: ";
    //cin >> setw(20);
    cin.getline(name,20); //cin.getline >> name;
    cout << "Hi " << name << endl;
    cout << "Press the ENTER key to end this program.";
    cin.get(go); // >> go;
    return 0;
}
```

2. Predict the output of the following function assuming the user enters “green tomatoes”

```
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    const int SIZE = 20;
    char userInput[SIZE];
    cout << "What is your name? ";
    cin.getline(userInput, SIZE);
    cout << "Hello " << userInput << endl;
    cout << "Your name is less than " << SIZE - 1 << " characters.";
```

SOLN:

```
What is your name? green tomatoes
Hello green tomatoes
Your name is less than 19 characters.
```

3. Consider the following function definition (numbers in the margin are there for reference.)

```

10 void randMatrix(double matrix[rows][cols]) {
11     const int MAXINT = 100000;
12     double sum = 0;
13     for(int i = 0; i < rows; ++i) {
14         sum = 0;
15         for(int j = 0; j < cols - 1; ++j) {
16             matrix[i][j] = (double) (rand()) / (double) ((j+1) * MAXINT);
17             sum += matrix[i][j];
18         } // end for
19         matrix[i][cols-1] = 1-sum;
20     } // end for
21 } // end randMatrix

```

a. Are both the variables “rows” and “cols” necessary in the parameter list of the function?

SOLN: You need [cols] to be specified, but [rows] is optional – you could leave it unspecified as empty brackets: []. This is because memory chunks need to be of a fixed size, but you don’t necessarily say how many of these chunks you need. The size of memory chunks must be fixed, but not the number of chunks.

b. Would a syntax error be generated by replacing line 14 by “MAXINT = 1;”?

SOLN: Yes, MAXINT is declared and initialized as a constant integer, that means it cannot be changed. This is an illustration of the principle that the program design should not allow the programmer the ability to change things which should not be changed.

4. Consider the following code fragment:

```

for(int i = 0; i < rows; ++i)
{
    sum = 0;
    for(int j = 0; j < cols - 1; ++j)
    {
        matrix[i][j] = (double) (rand()) / (double) ((j+1) * MAXINT);
        sum += matrix[i][j];
        cout << sum;
    }
    cout << endl;
}

```

a. How would you declare matrix so that this for loop would work properly?

SOLN: Either `double matrix[rows][cols-1];` or `double matrix[][cols-1];` or any integer values for rows and cols which are larger.

b. Suppose that rows = cols = 3 and the rand() numbers in come out (miraculously) as multiples of 500: 500, 1000, 1500,...

What will the code print to the screen?

SOLN: A good way to keep track of what’s what is to tabulate the variable values as you thread through the program. The sum is initialized to 0 and the nested for loops count j from 0 to 1 for each count of i from 0, 1, 2. The “rand()” values are coming out in multiples of 500 and sum is accumulating the values of matrix which rand() divided by the product of MAXINT (100000) with one more than j. Note that sum is reset to zero each time we exit the inner for loop.

MAXINT	sum	i	j	matrix[i][j]	rand()
100000	0.005	0	0	0.005	500
	0.01		1	0.005	1000
	0.015	1	0	0.015	1500
	0.025		1	0.01	2000
	0.025	2	0	0.025	2500
	0.04		1	0.015	3000

However, there's no "endl" between the outputs in the inner loop so the two outputs will be concatenated on one line as shown at right:

```
0.0050.01
0.0150.025
0.0250.04
```

- c. How would the output change if the casts to (double) were removed?
 SOLN: If the (double) type modifier is removed then the outputs would all be zero, two on each line, as shown at right:

```
00
00
00
```

5. Consider the following code fragment:

```
for(int i = 4; i > 0; --i)
{
    cout << (i * i) % (i + 1);
}
```

- a. Write a do-while loop to accomplish the same task.

SOLN:

```
int i = 4;
do
{
    cout << (i * i) % (i + 1);
    --i;
} while (i > 0);
```

- b. What is output to the screen by this fragment?

SOLN: Since $16 \bmod 5 = 9 \bmod 4 = 4 \bmod 3 = 1 \bmod 2 = 1$, and there are no space or new line characters, the output is just four adjacent 1's: 1111.

6. Write code using the if/else structure that is equivalent to the following:

```
switch(n)
{
    case 1 : x += 5;
           break;
    case 2 : x *= 5;
           break;
    default : x -= 5;
            break;
}
```

SOLN: **if**(n==1) x += 5; **else if**(n==2) x *= 5; **else** x -= 5;

7. Given that the value of the integer variable sum is 0 and the value of the integer variable prod is 10, identify each of the following Boolean expressions as either true or false.

- $(\text{sum} \neq 0) \ \&\& \ (\text{prod} < 10)$ is false.
- $(\text{sum} == 0) \ \&\& \ (\text{prod} < 20)$ is true.
- $(\text{sum} < 5) \ || \ (\text{prod} > 20)$ is true.
- $!(\text{prod} == 10)$ is false.
- $(\text{sum} == 1) \ \&\& \ (\text{sum} < \text{prod})$ is false.
- $(\text{sum} < 10) \ || \ (\text{prod} < \text{sum})$ is true.
- $!(\text{sum} < 10) \ || \ (\text{sum} < \text{prod}) \ \&\& \ (\text{sum} \geq 0)$ is true. (bit tricky since you need to know that $\&\&$ has precedence over $||$.)
- $((\text{sum}/\text{prod}) \geq 0) \ || \ (\text{sum} < 20)$ is true.
- $(\text{prod} < 20) \ \&\& \ (\text{prod}/(++\text{sum}) > 7)$ is true.
- $(\text{prod}/(\text{sum}++) > 7) \ \&\& \ (\text{prod} > 0)$ is true. (both pre-increment and post-increment have precedence over the division operator.)
- $(5 \ \&\& \ 7) + (!6)$ is true.

8. Suppose Hamlet.txt contains the text of Hamlet and is read by the following program:

```
int main() {
    const int ARRAY_SIZE = 200;    // Array size
    char junkWord[20];             // String
    char words[ARRAY_SIZE][20];    // Array of strings
    int count, i = 0;              // Loop counter variables
    ifstream inputFile;            // Input file stream object
    inputFile.open("Hamlet.txt");  // Open the file.
    // Read the words from the file into an array.
    for (count = 1; count < 20000; ++count) {
        if(count%100==0) {
            inputFile >> words[i];
            ++i;
        }
        else inputFile >> junkWord;
    }
    // Close the file.
    inputFile.close();
    // Display the words:
    cout << "The words are: ";
    for (count = 0; count < ARRAY_SIZE; count++)
        cout << words[count] << " ";
    return 0;
}
```

Here is the output of the program:

```
The words are: Volunteers is wish fifty Etext replace later on might not, copyri
ght other PUNITIVE opportunity harmless this converted the Carnegie-Mellon Hamle
t. good 'tis When of art watch to the sort, precurse in a dare in remembrance fr
ame, good That Pol. colour of impious throw thee that I the see follow'd The to
up My was twelve, Do but voice liquid double Of own 'Tis his Pol. for air. my co
mplexion, health of be so Mar. I and apt; have soft! tetter thy Yea, It Hor. my
but sword: disposition poor Of lord. must you, with me, his Long Whose age nor w
ithin the liege, The Was down. brief:--your comes Doubt man thy That her Honest,
far potently meeting both? what O Ros. justly you to neither, humorous Are will
Ay, my man be God a of me carbuncles, sword, than is all,-- their would from I,
ere They this.--the him him. frankly loneliness.--We to the cowards never for m
yself the face, all on us Speak to was Christian, and dear Hast very [Danish It
thought life pours counsel; round aught, me we makes day Madam, it drugs me A in
right stonish can Believe mystery; camel such range. many. And
```

a. Describe what the program does.

SOLN: The program creates an input file stream from Hamlet.txt and reads through the first 19999 words, saving every hundredth word into an array of 199 words called words. It then displays these words on the console, each word separated with a blank from the next.

b. What is the purpose of the junkWord array?

SOLN: This array is just a way of skipping over words until the word count is a multiple of a hundred.

c. How would you modify the program to read every 20th word into the array words.

SOLN: Instead of "if(count%100==0)", put "if(count%20==0)".

d. Write a function prototype and definition for a function called readWords() that fills up the words array in the same manner, but simplifies the body of main() so that handling of reading is accomplished by a call to readWords().

SOLN: You'd want readWords() to take at least the array of words and its length:

```
void readWords(char [][][20], const int);
```

9. Write a program that asks a user to enter two numbers and then outputs the sum of those numbers.

SOLN:

```
#include <iostream>
using namespace std;
int main() {
    int a, b;
    cout << "Enter two integers: ";
    cin >> a >> b;
    cout << a+b;
    cin.ignore(); cin.get();
}
```

10. Write a program that asks a user to enter two numbers and inputs those two numbers as the first two elements of an array called "foo." Then write code that uses a for loop to fill the next 8 elements of that array with numbers such that each element beyond the second is the sum of the previous two elements. So, for example, if the user enters 5 and 6, then the array contains the elements {5, 6, 11, 17, 28, 45, 73, 118, 191, 309} .

SOLN:

```
#include <iostream>
using namespace std;

int main() {
    int a[10];
    cout << "Input two numbers to Fibonacci: ";
    cin >> a[0] >> a[1];
    for(int i = 2; i < 11; i++)
        a[i] = a[i-1] + a[i-2];
    cout << "\nThe series is ";
    for(int i = 0; i < 11; ++i)
        cout << a[i] << " ";
    cin.ignore(); cin.get();
}
```