Physics 5 – Final Exam – Spring 2011 Name_____

Write your answers to the following questions on separate paper. Do not use an electronic computer.

1. Write a program that loops indefinitely. In each iteration of the loop, read in an integer N (declared as an `int`) that is inputted by a user, output N/5 if N is nonnegative and divisible by 5, and -1 otherwise. Use the ternary operator (`?:`) to accomplish this. (Hint: the modulus operator may be useful.)

2. Modify the code from #1 above so that if the condition fails, nothing is printed. Use an `if` and a `continue` command (instead of the ternary operator) to accomplish this.

3. Consider the following program (assume `<iostream>` is included and "`using namespace std`".)

```
1  int accumulator = 0, sam, pam;
2  cout << "\nEnter integers for sam and pam: ";
3  cin >> sam >> pam;
4  while (true ) {
5     if (pam == 0) break ;
6     accumulator += ((pam % 2 == 1) ? sam : 0);
7     pam /= 2;
8     sam *= 2;
9  }
10 cout << accumulator << "\n";
```

   a. Complete the following tables until the program completes, or indicate that it's an infinite loop:

sam	pam	accumulator
5	4	0
⋮	⋮	⋮

sam	pam	accumulator
6	17	0
⋮	⋮	⋮

   b. In a few words, describe the output of this program in terms of the inputted values for `sam` and `pam`, and how it works.

4. Write a C++ program that will take a list of `N` integers, find its mean (as a double), maximum value, minimum value, and range (the difference between the max and the min). Your program will first ask for N, the number of integers in the list, which the user will input. Then the user will input `N` more numbers. Design the program to call functions for evaluating the mean, max and min.
   Here is a sample input sequence:
   `3 <--N 2 1 3`
   Three numbers are inputted: 2, 1, 3. The output should be as follows:
   Mean: 2 Max: 3 Min: 1 Range: 2

5. What does the following program snippet do? The line numbers are obviously not part of the program – you can use them for reference in your description.

```
1  // N is a nonnegative integer
2  double acc = 0;
3  for (int i = 1; i <= N; ++i)
4  {
5    double term = (1.0/i);
6    acc += term * term;
7    for (int j= 1; j<i; ++j)
```

```
8    {
9       acc *= -1;
10   }
11 }
12 cout << acc << "\n";
```

6. Assume that the array x has been initialized as
   `float x[100] = {2.2, 4.4, 6.6, 8.8};`
   The following code snippet is supposed to insert the float t so the resulting array will be in increasing order, but it has an error. How would you fix it?

```
void insert(float x[], int n, float t)
{ for (int i=n; i>0 && x[i-1] > t; i--)
    x[i] = x[i-1];  // shift larger elements up
  x[i] = t;
}
```

7. What is output by the following code? (Function definitions are on the next page.)

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

void rot13(string& s);

int main()
{ string s = "ABcd$Mn";
  cout << "s = " << s << endl;
  rot13(s);
  cout << "s = " << s << endl;
}

bool isUpCaseBot(char c)
{ return bool(c >= 'A' && c <= 'M');
}

bool isUpCaseTop(char c)
{ return bool(c >= 'N' && c <= 'Z');
}

bool isLowCaseBot(char c)
{ return bool(c >= 'a' && c <= 'm');
}

bool isLowCaseTop(char c)
{ return bool(c >= 'n' && c <= 'z');
}

void rot13(string& s) {
      int i = 0;
      while(s[i] != '\0') {
            if(isUpCaseBot(s[i]) || isLowCaseBot(s[i])) s[i]+=13;
            else if(isUpCaseTop(s[i]) || isLowCaseTop(s[i])) s[i]-=13;
            ++i;
      }
}
```

8. Consider the following complete C++ program:

```
#include <iostream>
#include <ctime>
#include <climits>

const int MAX = INT_MAX;

using namespace std;

class Random
{ public:
    Random(unsigned long seed=0);
    void seed(unsigned long seed=0);   // allows client to reset _seed
    int integer(unsigned long hi=MAX, unsigned long lo=1);
    double real();
  private:
    unsigned long _seed; // INVARIANT: 0 <= _seed <= unsigned long_MAX
    void _randomize();   // resets _seed
};

Random::Random(unsigned long s)
{ if (s > 0) _seed = s;
  else _seed = time(NULL);
  _randomize();
}

void Random::seed(unsigned long s)
{ if (s > 0) _seed = s;
  else _seed = time(NULL);
  _randomize();
}

int Random::integer(unsigned long hi, unsigned long lo)
{ _randomize();
  return _seed/10 % (hi - lo + 1) + lo;
}

double Random::real()
{ _randomize();
  return double(_seed)/ULONG_MAX;
}

void Random::_randomize()
{ _seed = (1103515245*_seed + 123456789) % ULONG_MAX;
}

class Dice {
  public:
    Dice() : _sum(3) { }
    void toss()
    { _sum = _random.integer(6) + _random.integer(6) + _random.integer(6);
    }
    int sum() { return _sum; }
  private:
    int _sum;   // the sum of the three dice
    Random _random;  // random number generator
};


int main() {
      Dice d1;
```

```
      for(int i = 1; i < 10; ++i) {
            d1.toss();
            cout << d1.sum() << endl;
      }
      cin.ignore();
      return 0;
}
```

a. Why are the headers `<ctime>` and `<climits>` included?
b. Describe what the function `integer()` work? Explain the inputs and output of this function.
c. How does `Dice` use `Random`?
d. How does the constructor in Dice() work?
e. Describe what the for loop in main() does. Give as much detail as is appropriate.
f. How would you modify main() to test that Dice accurately simulates rolling three dice?

9. Suppose that p = 0x0012FF50, &a = 0x0012FF44, &p = 0x0012FF38.
What will be output by the following code?

```
int main()
{ int a[] = { 22, 33, 44, 55, 66, 77, 88, 99 };
  int* p = &a[3];                              // p points to a[3]
  cout << "p = " << p << ", *p = " << *p;
  cout << "\n&a = " << &a;                     // ok: a is an lvalue
  cout << "\n&p = " << &p;                     // ok: p is an lvalue
  cout << "\n&(a[5]) = " << &(a[5]);       // ok: a[5] is an lvalue
  cout << "\n&(*(a+5)) = " << &(*(a+5));// ok: *(a+5) is an lvalue
  cout << "\n&(*(p+2)) = " << &(*(p+2));// ok: *(p+2) is an lvalue
  cout << "\n&(p+2) = " << p+2; //&(p+2);  // ERROR: p+2 is not an lvalue
  cout << endl;
}
```

10. What's the most important thing you learned in this class? Any answer (not blank) will do.\

Physics 5 – Final Exam Solutions – Spring 2011 .

1. Write a program that loops indefinitely. In each iteration of the loop, read in an integer N (declared as an `int`) that is inputted by a user, output N/5 if N is nonnegative and divisible by 5, and -1 otherwise. Use the ternary operator (`?:`) to accomplish this. (Hint: the modulus operator may be useful.)

SOLN:

```
#include <iostream>

using namespace std;

int main() {
```

```
      int N;
      while(1) {
            cin >> N;
            (N>=0 && N%5 == 0) ? cout << N/5 << endl : cout << -1 << endl;
      }
}
```

2. Modify the code from #1 above so that if the condition fails, nothing is printed. Use an `if` and a `continue` command (instead of the ternary operator) to accomplish this.

   SOLN:

```
#include <iostream>

using namespace std;

int main() {
      int N;
      while(1) {
            cin >> N;
            if (N>=0 && N%5 == 0) cout << N/5 << endl;
            else cout << -1 << endl;
            continue;
      }
}
```

3. Consider the following program (assume `<iostream>` is included and "`using namespace std`".)

```
1  int accumulator = 0, sam, pam;
2  cout << "\nEnter integers for sam and pam: ";
3  cin >> sam >> pam;
4  while (true )
5  {
6     if (pam == 0) break ;
7     accumulator += ((pam % 2 == 1) ? sam : 0);
8     pam /= 2;
9     sam *= 2;
10 }
11 cout << accumulator << "\n";
```

   a. Complete the following tables until the program completes, or indicate that it's an infinite loop:

accumulator	pam	sam
0	2	5
0	1	10
20	0	20

accumulator	sam	pam
0	6	17
6	12	8
6	24	4
6	48	2
102	96	1
102	192	0

b. In a few words, describe the output of this program in terms of the inputted values for `sam` and `pam`, and how it works.
SOLN: What happening is that the product of sam and pam is being computed by multiplying pam by the sum of sam's binary decomposition, in say 5*4 = 5*(2^2) and in the case of 6*17 = 6(1+2^4)

4. Write a C++ program that will take a list of `N` integers, find its mean (as a double), maximum value, minimum value, and range (the difference between the max and the min). Your program will first ask for N, the number of integers in the list, which the user will input. Then the user will input `N` more numbers. Design the program to call functions for evaluating the mean, max and min.

Here is a sample input sequence:
`3 <--N 2 1 3`
Three numbers are inputted: 2, 1, 3. The output should be as follows:
Mean: 2 Max: 3 Min: 1 Range: 2
SOLN:

```
#include <iostream>

using namespace std;

int main() {
      int N, min = 0, max = 0, sum = 0, x, cntr;
      cout << "\nEnter the number of integers you will enter: ";
      cin >> N; cntr = N-1;
      cout << "\nEnter " << N << " numbers separated by spaces: ";
      cin >> x;
      min =x; max = x; sum =x;
      while(cntr>0) {
            cntr--;
            cin >> x;
            if (x>max) max = x;
            if (x<min) min = x;
            sum += x;
      }
      cout << "Mean: " << ((double)sum/(double)N) << " "
            << "max: " << max << " "
            << "min: " << min << " "
            << "range: " << max - min << endl;
}
```

oops…that didn't use functions, did it? How about this:

```
#include <iostream>

using namespace std;

float mean(int [], int);
int min(int [], int);
int max(int [], int);

int main() {
      int nums[100];
      int N;
      cout << "\nInput the number of integers in your list: ";
      cin >> N;
      cout << "\nNow input the numbers, separated by spaces: ";
      for(int i = 0; i < N; i++)
            cin >> nums[i];
      cout << "\nMean: " << mean(nums, N)
             << " Max: " << max(nums, N)
             << " Min: " << min(nums, N)
             << " Range: " << max(nums, N) - min(nums, N) << endl;
      return 0;
}

float mean(int nums[], int N) {
      float sum = 0;
      for(int i = 0; i < N; ++i) sum += nums[i];
      return sum/N;
}

int max(int nums[], int N) {
      int MAX = nums[0];
      for(int i = 1; i < N; i++)
            if (nums[i] > MAX) MAX = nums[i];
      return MAX;
}

int min(int nums[], int N) {
      int MIN = nums[0];
      for(int i = 1; i < N; i++)
            if (nums[i] < MIN) MIN = nums[i];
      return MIN;
}
```

5. What does the following program snippet do? The line numbers are obviously not part of the program – you can use them for reference in your description.

```
1  // N is a nonnegative integer
2  double acc = 0;
3  for (int i = 1; i <= N; ++i)
4  {
5    double term = (1.0/i);
6    acc += term * term;
7    for (int j= 1; j<i; ++j)
8    {
9       acc *= -1;
10   }
11 }
12 cout << acc << "\n";
```

SOLN: Consider the case where $N = 1$. In this case the outer for loop executes exactly once and the inner loop does not execute at all. The variable term is 1 and so is acc and so the number 1 is printed. Now suppose $N = 2$. In this case, the outer for loop cycles twice but the inner loop executes only once. When i = 2, term is set to 0.5, acc increases to 1.25 and is then negated to its final value, -1.25.
When $N = 3$, 1/9 is added to acc, so it's new value is -1.25 + 0.11111 = -1.13889, which is negated twice, so it's final value is -1.13889
When $N = 4$, 1/16 is added to the running total giving acc = -1.13889 + 0.0625 = - 1.07639. This is then negated three times to yield a final value of acc = 1.07639.
When $N = 5$, 1/25 is added and the value is negated four times to yield 1.11639
When$N = 6$, 1/36 is added and the value is negated five times to yield -1.14417
The pattern boes like this: $-\left(-\left(-\left(-\left(1+\frac{1}{4}\right)+\frac{1}{9}+\frac{1}{16}\right)+\frac{1}{25}+\frac{1}{36}\right)+\frac{1}{49}+\frac{1}{64}\right)+\cdots$

6. Assume that the array x has been initialized as
```
float x[100] = {2.2, 4.4, 6.6, 8.8};
```
The following code snippet is supposed to insert the float t so the resulting array will be in increasing order, but it has an error. How would you fix it?

```
void insert(float x[], int n, float t)
{ for (int i=n; i>0 && x[i-1] > t; i--)
    x[i] = x[i-1];  // shift larger elements up
  x[i] = t;
}
```

SOLN: The error is on the next to last line: `x[i] = t;` This can fixed by declaring i to be a variable whose scope extends beyond the for loop, like so:

```
void insert(float x[], int n, float t)
{ int i;
  for (int i=n; i>0 && x[i-1] > t; i--)
    x[i] = x[i-1];  // shift larger elements up
  x[i] = t;
}
```

7. What is output by the following code? (Function definitions are on the next page.)

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

void rot13(string& s);

int main()
{ string s = "ABcd$Mn";
  cout << "s = " << s << endl;
  rot13(s);
  cout << "s = " << s << endl;
}

bool isUpCaseBot(char c)
{ return bool(c >= 'A' && c <= 'M');
}

bool isUpCaseTop(char c)
{ return bool(c >= 'N' && c <= 'Z');
}
```

```
bool isLowCaseBot(char c)
{ return bool(c >= 'a' && c <= 'm');
}

bool isLowCaseTop(char c)
{ return bool(c >= 'n' && c <= 'z');
}

void rot13(string& s) {
      int i = 0;
      while(s[i] != '\0') {
            if(isUpCaseBot(s[i]) || isLowCaseBot(s[i])) s[i]+=13;
            else if(isUpCaseTop(s[i]) || isLowCaseTop(s[i])) s[i]-=13;
            ++i;
      }
}
```

SOLN:

The code takes all alphabetic characters (upper and lower cases) and rotates them 13 characters through the alphabet, preserving the case. So the output is

```
s = ABcd$Mn
s = NOpq$Za
```

8. Consider the following complete C++ program:

```
#include <iostream>
#include <ctime>
#include <climits>

const int MAX = INT_MAX;

using namespace std;

class Random
{ public:
    Random(unsigned long seed=0);
    void seed(unsigned long seed=0);   // allows client to reset _seed
    int integer(unsigned long hi=MAX, unsigned long lo=1);
    double real();
  private:
    unsigned long _seed; // INVARIANT: 0 <= _seed <= unsigned long_MAX
    void _randomize();   // resets _seed
};

Random::Random(unsigned long s)
{ if (s > 0) _seed = s;
  else _seed = time(NULL);
  _randomize();
}

void Random::seed(unsigned long s)
{ if (s > 0) _seed = s;
  else _seed = time(NULL);
  _randomize();
}

int Random::integer(unsigned long hi, unsigned long lo)
{ _randomize();
  return _seed/10 % (hi - lo + 1) + lo;
}
```

```
double Random::real()
{ _randomize();
  return double(_seed)/ULONG_MAX;
}

void Random::_randomize()
{ _seed = (1103515245*_seed + 123456789) % ULONG_MAX;
}

class Dice {
  public:
    Dice() : _sum(3) { }
    void toss()
    { _sum = _random.integer(6) + _random.integer(6) + _random.integer(6);
    }
    int sum() { return _sum; }
  private:
    int _sum;   // the sum of the three dice
    Random _random;  // random number generator
};


int main() {
      Dice d1;
      for(int i = 1; i < 10; ++i) {
            d1.toss();
            cout << d1.sum() << endl;
      }
      cin.ignore();
      return 0;
}
```

a. Why are the headers `<ctime>` and `<climits>` included?
   SOLN: The first is for the random seed and the second or `INT_MAX` and `ULONG_MAX.`

b. Describe what the function `integer()` work? Explain the inputs and output of this function.
   SOLN: `integer()` takes two default arguments, `hi`, with a default value of INT_MAX from `climits` and `lo` with a default value of `1`.
   It then calls the private method `_randomize()` of `Dice` to set the `_seed` in terms of itself like so:
   `_seed = (1103515245*_seed + 123456789) % ULONG_MAX;`
   and then `integer()` returns `_seed/10 % (hi - lo + 1) + lo;` This a pseudo-random number between `lo` and `hi`.

c. How does `Dice` use `Random`?
   SOLN: `Dice` uses `Random` as a random number generator for its private data field, `_random`. This, in turn, is used by the method `toss()` to simulate the tossing of three dice.

d. How does the constructor `Dice()` work?
   SOLN: `Dice() : _sum(3) { }` uses an initialization list to set the default value for `_sum` to 3.

e. Describe what the `for` loop in `main()` does. Give as much detail as is appropriate.
   SOLN: The `for` loop executes the toss() function of Dice d1 9 times and displays the sum of the three virtual dice with a line feed in between each.

f. How would you modify `main()` to test that Dice accurately simulates rolling three dice?
   SOLN: Do, say, 100000 tosses and record the relative frequency of each toss – does this compare favorably with the theoretical probability for each?

9. Suppose that p = 0x0012FF50, &a = 0x0012FF44, &p = 0x0012FF38.
What will be output by the following code?

```
int main()
{ int a[] = { 22, 33, 44, 55, 66, 77, 88, 99 };
  int* p = &a[3];                              // p points to a[3]
  cout << "p = " << p << ", *p = " << *p;
  cout << "\n&a = " << &a;                     // ok: a is an lvalue
  cout << "\n&p = " << &p;                     // ok: p is an lvalue
  cout << "\n&(a[5]) = " << &(a[5]);       // ok: a[5] is an lvalue
  cout << "\n&(*(a+5)) = " << &(*(a+5));// ok: *(a+5) is an lvalue
  cout << "\n&(*(p+2)) = " << &(*(p+2));// ok: *(p+2) is an lvalue
  cout << "\n&(p+2) = " << p+2; //&(p+2);  // ERROR: p+2 is not an lvalue
  cout << endl;
}
```

SOLN:

```
p = 0012FF50, *p = 55
&a = 0012FF44
&p = 0012FF38
&(a[5]) = 0012FF58
&(*(a+5)) = 0012FF58
&(*(p+2)) = 0012FF58
&(p+2) = 0012FF58
```

10. What's the most important thing you learned in this class? Any answer (not blank) will do.
"I think the most important thing I learned in this class is that there are so many ways to accomplish the same task when it comes to programming. Every time I thought of an algorithm to solve my problem, I eventually found an easier way or more efficient way of doing the same thing. It really goes to show how powerful this language can be, but also how many problems it can create given the amount of code that exists! I spend more time learning and debugging syntax than I do writing the code!"

"I learned how to override math operators and create my own math functions."

"The most important thing I learned in this class was how to read code, and the funnest thing was how to make video games, or make a house I could put in a video game. I had fun!"

"How to ask for help!"

"The most important thing I learned is that you need to have a strong basis on C++ and its functions. Learning something DarkGDK is just like learning how to use a specialized code made from the C++ basis. Also, you need to think outside of the box for some of these codes…"

"The most important thing I learned in this class was that computers do not interpret data. They simply carry out user-supplied commands input via a program written in a programming language. Effectively, computers do not think for themselves. This seems obvious but I found it interesting to examine a computer's ability to carry out my commands with great speed. If something did not work, it was my fault. I input something incorrectly. A human can not understand someone speaking in an unfamiliar language just like a computer cannot understand user commands input incorrectly."

"I learned that C++ is an object-oriented programming language that has extreme potential. Although I didn't understand most of what was going on, the text book seemed to help a little. The quizzes helped too. Crazy triangles is one crazy project."

“Mostly about debugging with cout, putting and assigning variables, what are its types & it application to find the are, volume, etc.”

“This is really [the most] interest[ing] class I have taken this semester. I have learned a lot of thing[s] about programming in this class. The most interest[ing] program I liked is root chao[s]. This program taught me how to combine the differences code to make a program fun. Beside that, my professor have a lot of experiences about the programming, he have me a lot of interesting program which I think I may never learn it with other class. At the end of my words, I would like to say thank and appreciate my professor who has hand out his knowledge for me.”

“Understanding machine language, math-equation solving, creativity in instructing tasks for a computer and resolving complex problems through a single program.”

“It’s difficult to pinpoint what the most important thing to me was this semester. There was so many things going on, but I would have to say it is the endless possibilities with C++ and computer programming as a whole. There are so many different ways to program and not one rule set in stone, or one specific method. On to the next one! ☺.”

“The first syntax error to look for usually deals with “i” or “{“ “}”.

“Being my first class ever to do with any type of computer code, to be honest I learned many individual important things. I suppose however, the conversion from binary to digit (sic) and vice versa was important because in any area of computing I think that knowledge in that area of zeros and 1’s is important. I had a really hard time in this class but being able to output solution by inputting commands, connecting to DarkGDK, etc. was all important in getting a foundation to later work with more complex code.”

“Learned the basic idea of how coding operates however also realized in order to be successful in it, you had to understand high level math.”

“The most important thing in class were the class functions. If class functions were created more into depth in order to solve or figure out how the unsolved proposed equations of mathematics probably can be solved. I’m not sure if it’s possible for a machine to solve this or it won’t. People work together to overcome many obstacles in programming.”

“I learned that I either must change my major or study more. The last math class I took and successfully passed was in ’08. I need to brush up on my math skills. Next class you teach it would be better to assume you were dealing with people that had never coded before. [fatuous praise delted]…I just wish the class would’ve been longer for me to fully grasp some of the concepts that I did not completely understand.”