CISC 181 section 080, Midterm 2 11/21/05

Name____

Please circle your section number:

Answer the multiple choice questions on a “Scantron Form”

Bubble in ONLY your Unix userid and your answers

DO NOT bubble in your id number or section
If you bubble in your SSN, the computer will **reject your form!!!**

Answer the remaining questions directly on the exam paper.

General Instructions

- The exam is 16% multiple choice, and 84% programming/short answer.
- The programming questions start with number 9. You may want to tackle them first, since they may take more time.
- DO NOT WRITE YOUR NAME ON ANY PAGE EXCEPT THIS ONE!
- You have 75 minutes. **Pace yourself,** and pay attention to the point values.
- Read *all* the directions *carefully* on each problem.
- Good luck.

Questions 1 through 4 deal with the following code excerpt:

```
// foo.cc

#include <iostream>
using namespace std;

int main(void)
{
   char mascot[10]="Blue Hens";
   char hometown[20]="Newark";
   char school[30]="University of Delaware";
   char state[20]="Delaware";

   char *a = mascot;
   char * const c = hometown;
   const char *b = school;
   const char * const d = state;


   cout << a << endl;
   cout << b << endl;
   cout << c << endl;
   cout << d << endl;

   // where some extra code might go

#include "extra.h"

   cout << a << endl;
   cout << b << endl;
   cout << c << endl;
   cout << d << endl;

}
```

1. (2 pts) One of these lines of code would produce a compiler error if placed at the spot indicated by `// where some extra code might go` ?
   Which one is WRONG?

   (a) `a[0] = ’F’;`

   (b) `b = hometown;`

   (c) `a[1] = ’r’;`

   (d) `b[0] = ’D’;`

   (e) `a = hometown;`

2. (2 pts) One of these lines of code would produce a compiler error if placed at the spot indicated by `// where some extra code might go` ?
   Which one is WRONG?

   (a) `a = mascot;`

   (b) `a = state;`

   (c) `b[1] = ’u’;`

   (d) `c[1] = ’m’;`

   (e) `a[1] = ’u’;`

3. (2 pts) One of these lines of code would produce a compiler error if placed at the spot indicated by `// where some extra code might go` ?
   Which one is WRONG?

   (a) `d = state;`

   (b) `b = hometown;`

   (c) `c[0] = ’D’;`

   (d) `b = mascot`

   (e) `c[1] = ’i’;`

4. (2 pts) Read carefully! Only one of these lines of code would NOT produce a compiler error if placed at the spot indicated by `// where some extra code might go` ?
   That is, which one is CORRECT?

   (a) `d[1] = ’u’;`

   (b) `c[1] = ’r’;`

   (c) `c = d;`

   (d) `c = a;`

   (e) `d = hometown;`

Questions 5 through 8 refer to the fragment of code shown in Figure 1 that comes from a complete C++ program.

Figure 1: Declaration of a struct

```
...
struct Node
{
  int data;
  Node *next;
};

int main(void)
{
  Node *p;
  Node s;
  p = new Node;
  ...
}
```

5. (2 pts) Only one of the following lines could legally appear in the program referred to in Figure 1, at or after the line indicated by `...`
   Which one?

   (a) `data = 3`

   (b) `s.data = 3;`

   (c) `(*s).data = 3;`

   (d) `s->data = 3;`

   (e) `p.data = 3;`

6. (2 pts) Only one of the following lines could legally appear in the program referred to in Figure 1, at or after the line indicated by `...`
   Which one?

   (a) `next = NULL;`

   (b) `p->next = NULL;`

   (c) `s->next = NULL;`

   (d) `p.next = NULL;`

   (e) `(*s).next = NULL;`

7. (2 pts) Which of the following is equivalent to `p->data`?

   (a) `(*p).data`

   (b) `*(p.data)`

   (c) `p.(*data)`

   (d) `p[data]`

   (e) `p[1].data`

8. (2 pts) Which of the following is equivalent to `p->next`? (Hint: This is is tricky: think carefully about the equivalence of array subscripts and dereferencing of pointers.)

   (a) `*(p.next)`

   (b) `p[next]`

   (c) `p[0].next`

   (d) `p[1].next`

   (e) `*p[next]`

9. (24 pts) The main program listed below demonstrates a function that can put an array of temperature values in order from hottest to coldest. Sample output is given, along with a complete main program, and three function definitions. The last two function definitions on the following page are incomplete. Finish them, using the hints provided. Be extra careful with the order of parameters.

   Sample output:

```
> ./a.out
Temps:  45.0  32.0  54.0   8.5  19.1
Hottest to Coldest:  54.0  45.0  32.0  19.1   8.5
>
```

   Code:

```
/* Esmerelda Weatherwax,   CISC181, Fall 2004  TA:Jason Ogg     */
/* Demonstrate call to function that puts temperatures in order */
/* from hottest to coldest                                      */

#include <iostream>
using namespace std;

void hottestToColdest(int howMany, double temps[]);
void putColdestAtEnd(double tempArray[], int numTemps);
void outputArrayOnOneLine(int count, double array[]);

int main(void)
{
  double tempReadings[5]={45.0, 32.0, 54.0, 8.5, 19.1};

  /* output the temperatures before the function call */

  cout << "Temps: ";
  outputArrayOnOneLine(5, tempReadings);
  cout << endl;

  /* put temperatures in order */

  hottestToColdest(5, tempReadings);


  /* output the temperatures after the function call */

  cout << "Hottest to Coldest: ";
  outputArrayOnOneLine(5, tempReadings);
  cout << endl;

  return 0;

}
```

```
void outputArrayOnOneLine(int count, double array[])
{
  int i;
  cout << array[0];
  for (i=1; i<count; i++)
    cout << " " << array[i];

}

void hottestToColdest(int howMany, double temps[])
{
    /* Hints: use a for loop, and a call to putColdestAtEnd.

     For you to decide: does the call go inside or outside
     the for loop?   Does the loop count up or down?
     What do you pass to putColdestAtEnd?  Does it change
     or stay the same each time through the loop?

     About 3-4 lines of code is all you need.
  */






}

void putColdestAtEnd(double tempArray[], int numTemps)
{

  /* Hint: use a loop to find the index of coldest temperature,
     then swap that element with the last one in the array

     About 6-8 lines of code will be enough. */

  int j;
  double temp;
  int indexOfColdest = 0; /* index of coldest element */













}
```

10. (24 pts) Write the .h file only (the class declaration) for a class that would represent a drawing of the type that you did in project 1.

    You should include appropriate private data members, and function prototypes for the member functions. You do NOT need to implement the member functions for this exam.

    You should use the following representation for a drawing:

    - Keep track of the head and tail of a linked list of nodes
    - Each node will contain the x and y values for a point, represented as doubles.
    - Each node will also contain a bool indicating whether to lift the pen before proceeding to the next point.

    For example, the letter X might be represented by a linked list of nodes, containing:

    - x=0, y=0, lift=false
    - x=4, y=8, lift=true
    - x=0, y=8, lift=false
    - x=4, y=0, lift=true

    Include prototypes for the following member functions:

    - a default constructor to create an empty drawing
    - a function to add a node at the end of the linked list
    - a function to draw the drawing (by outputting gnuplot commands to an ostream object passed by reference).
    - a copy constructor
    - a destructor

    Note that a proper implementation that has a copy constructor and a destructor would also require an overloaded assignment operator, however I will not ask you for that on this exam; I'll ask for that on the final exam.

    Be sure to use `const` on any and all member functions for which it is appropriate.

    There is extra space for your answer on the next page.

Extra space for your answer

The following questions deal with a code excerpt that appears on the last page of the exam, with a comment:

`// Code for Type Expressions Questions`

11. (20 pts) Suppose you run the C++ program on the previous page, and the program reaches the line of code that prints out `Hi`.

Indicate whether, at that point in time, the expression given in the table on the left

- refers to a memory location on the *stack*,
- refers to a memory location on the *heap*,
- would cause an *error* (e.g. dereferencing something that isn't a pointer.)

Then, in the table on the right, fill in the type of each expression, or write "error" if the expression would not be valid (e.g. dereferencing something that isn't a pointer, or using the dot operator (`.`) on something that isn't a `struct`.

expression	*Circle stack, heap or error*
a	stack heap error
*a	stack heap error
b	stack heap error
*b	stack heap error
c	stack heap error
*c	stack heap error
*d	stack heap error
*e	stack heap error
f	stack heap error
*f	stack heap error
g	stack heap error
*g	stack heap error
p	stack heap error
*p	stack heap error
*q	stack heap error
*r	stack heap error
*s	stack heap error
*t	stack heap error

expression	*type*
a	int
*a	
&a	
b	
*b	
&b	
d	
*d	
e	
&e	
p	
*r	
&s	
t->x	
q.y	
q->y	
(*p).x	
&(p.x)	
argc	
argv[0]	
argv[0][1]	
argv	
&argc	

12. (16 pts) Draw a picture of the linked list that is constructed by the program `q2.cc` which appears on a separate page at the end of this exam.

Draw the state of what is in memory at the time the `return(0);` statement is reached.

In your picture, indicate where `head`, `tail`, and `p` are pointing. If you need more space, draw your picture next to the code on the following page.

End of Exam. Total Points: 100

XYAFRA91BDPE38A82D6DF3S

```
// q2.cc
// Example program for CISC181 Midterm Exam 2
// P. Conrad, Fall 2005, Section 080

#include <iostream>
using namespace std;

struct Node_S
{
  int data;
  Node_S *next;
};

int main(int argc, char *argv[])
{

  Node_S *head = NULL;
  Node_S *tail = NULL;
  Node_S *p = NULL;

  head = new Node_S;
  head->data = 4;
  head->next = NULL;
  tail = head;

  p = new Node_S;
  p->data = 9;
  p->next = NULL;
  tail -> next = p;
  tail = p;

  p = new Node_S;
  p->data = 2;
  p->next = NULL;
  tail -> next = p;
  tail = p;

  p = new Node_S;
  p->data = 1;
  p->next = head;
  head = p


  return 0;
} // end main
```

```
// Code for Type Expressions Questions
#include <iostream>
using namespace std;

struct Point_S
{
  int x;
  int y;
};

int main(int argc, char*argv[])



{
   int  a;
   int *b;
   int *c;

   double  d;
   double  e;
   double *f;
   double *g;

   Point_S *p;
   Point_S *q;
   Point_S *r;
   Point_S  s;
   Point_S  t;

   b = new int;
   c = &a;

   f = new double;
   g = &e;

   p = new Point_S;
   q = new Point_S;
   r = &s;

   cout << "Hi" << endl;
   return 0;
}
```

1. (d)
2. (c)
3. (a)
4. (b)
5. (b)
6. (b)
7. (a)
8. (c)
9.

```
/* Esmerelda Weatherwax,   CISC181, Fall 2004  TA:Jason Ogg     */
/* Demonstrate call to function that puts temperatures in order */
/* from hottest to coldest                                      */

#include <iostream>
using namespace std;

void hottestToColdest(int howMany, double temps[]);
void putColdestAtEnd(double tempArray[], int numTemps);
void outputArrayOnOneLine(int count, double array[]);

int main(void)
{
  double tempReadings[5]={45.0, 32.0, 54.0, 8.5, 19.1};

  /* output the temperatures before the function call */

  cout << "Temps: ";
  outputArrayOnOneLine(5, tempReadings);
  cout << endl;

  /* put temperatures in order */

  hottestToColdest(5, tempReadings);


  /* output the temperatures after the function call */

  cout << "Hottest to Coldest: ";
  outputArrayOnOneLine(5, tempReadings);
  cout << endl;

  return 0;

}

void outputArrayOnOneLine(int count, double array[])
{
  int i;
  cout << array[0];
  for (i=1; i<count; i++)
    cout << " " << array[i];

}

void hottestToColdest(int howMany, double temps[])
{
  /* Hints: use a for loop, and a call to putColdestAtEnd.

     For you to decide: does the call go inside or outside
     the for loop?   Does the loop count up or down?
     What do you pass to putColdestAtEnd?  Does it change
     or stay the same each time through the loop?
  */

  for (int i=howMany; i>1; i--)
    {
      putColdestAtEnd(temps, i);
    }

}

void putColdestAtEnd(double tempArray[], int numTemps)
{

  /* Hint: use a loop to find the index of coldest temperature,
     then swap that element with the last one in the array */

  int j;
  double temp;
  int indexOfColdest = 0; /* index of coldest element */




  for (j=1; j<numTemps; j++)
    {
      if (tempArray[j] < tempArray[indexOfColdest])
        {
          indexOfColdest = j;
        }

    }

  temp = tempArray[indexOfColdest];
  tempArray[indexOfColdest] = tempArray[numTemps - 1];
  tempArray[numTemps - 1] = temp;


}
```

10.

```

#include <iostream>

class Drawing_C
{

public:
   Drawing_C();
   Drawing_C(const Drawing_C &orig);
   ˜Drawing_C();
   void addPoint(double theX, double theY, bool theLift);
   void draw(std::ostream &out) const;

private:

  struct PointNode_S
  {
    double x;
    double y;
    bool lift;
    PointNode_S *next;
  };

  PointNode_S *head, *tail;
};
```

expression	*type*
a	int
*a	error
&a	int *
b	int*
*b	int
&b	int**
d	double
*d	error
e	double
&e	double *
p	Point_S *
*r	Point_S
&s	Point_S *
t->x	error
q.y	error
q->y	int
(*p).x	int
&(p.x)	error
argc	int
argv[0]	char *
argv[0][1]	char
argv	char **
&argc	int *

11.

expression	*Answer*
a	stack
*a	error
b	stack
*b	heap
c	stack
*c	stack
*d	error
*e	error
f	stack
*f	heap
g	stack
*g	stack
p	stack
*p	heap
*q	heap
*r	stack
*s	error
*t	error

12. head should point to the 1 node
1 should point to the 4 node
4 should point to the 9 node
9 should point to the 2 node
2 should point to null tail should point to the 2 node
p should point to the 1 node

End of Key, seed 3456 version A

Total Points: 0