INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

Department of Computer Science and Engineering

Kanwal Rekhi Building

CS101 Quiz

Sunday, 9 November 2014

[Please DO NOT turn this page till start of examination is announced]

Important:

1. There are 15 questions. You are expected to answer all questions.
2. For questions in '**Section 1**', please remember to write your answer on both **Section 1 answer sheets** supplied, immediately after attempting each question.
3. For questions in '**Section 2**', please remember to write you answer only in the **single answer sheet for section 2**.
4. Please do not try speculative guess work to avoid penalty of negative marks

Section 1 – Multiple Choice Questions

Write your answers on BOTH Section 1 – Answer Sheets

Q1. We are given the birthdate of children in the format MMYYDDZZ, where MM refers to the month (01 through 12), YYZZ refers to the year, and DD refers to the day (01 through 31). Note that the year YYZZ is split into two parts with DD in between. As an example, if a child was born on 9 November 2014, then the birthdate of the child in the above format would be 11200914. This is because MM is 11 (November), YY is 20 (since the year of birth is 2014), DD (the date of birth) is 09, and ZZ is 14 (since the year of birth is 2014).

You are required to write a function that accepts an integer representing a birthdate in the above format and returns an integer representing the year of birth (i.e. YYZZ).

The following is an incomplete definition of a function for achieving the above goal.

```
int findYearFromFormat(int formattedBDateTime) {
    int ZZ = E1;
    int YY = E2;
    return E3;
}
```

Indicate which option(s) for expressions E1, E2 and E3 will work for the above function:

A. E1: formattedBDateTime/100
   E2: (formattedBDateTime%10000)/100
   E3: 100*YY + ZZ

B. E1: formattedBDateTime/100
   E2: (formattedBDateTime%10000)/100
   E3: 100*ZZ + YY

C. E1: formattedBDateTime%100
   E2: (formattedBDateTime/10000)%100
   E3: 100*YY + ZZ

D. E1: formattedBDateTime%100
   E2: ((formattedBDateTime/10000)%100)*100
   E3: YY + ZZ

E. None of the above

Answer: C, D

Q2. Consider the following C++ program:

```
 #include <iostream>
 using namespace std;
 int main() {
   int a = 3;
   for ( ; ; ) {
     a *= 2;
     if (a > 100) {a -= 5; break;}
   }

   while (true) {
     a *= 2;
     if (a > 100) {a -= 5; break;}
   }

   do {
     a *= 2;
     if (a > 100) {a -= 5; break;}
   } while (true);

   cout << a << endl;
   return 0;
}
```

What is the value of 'a' printed by the above program?

A. 187

B. 369

C. 203

D. 733

E. None of the above

Answer: D

Q3. A programmer wants to write a variant of SelectionSort for sorting an array A of integers in ascending order. The programmer has written the following C++ function for this purpose:

```
void newSelectSort(int *A, int n) {
   int minEl, temp;

   for (int j = 0;  j < n; j++) {  // Begin for loop 1
     minEl = A[j];
     for (int i = j; i < n; i++) { // Begin for loop 2
        if (A[i] < minEl) { // Begin if
L1:        temp = A[i];
L2:        A[i] = minEl;
L3:        minEl = temp;
        } // End of if
     } // End of for loop 2
L4:  temp = A[j];
L5:  A[j] = minEl;
L6:  minEl = temp;
   } // End of for loop 1
}
```

It turns out that the above function correctly sorts the array A in ascending order. Interestingly, if some statements in the above function are deleted, it still continues to sort the array A in ascending order. Which statements in the above function can be deleted while still allowing the function 'newSelectSort' to sort the array A in ascending order?

A. L4 and L6

B. L2

C. L1 and L3

D. L5

E. None of the above

Answer: A (L4 and L6)

Q4. Consider the following C++ program:

```
1:  #include <iostream>
2:  using namespace std;
3:
4:  struct myStruct1 {
5:   int c;
6:   int getC() {return c;}
7:  };
8:
9:  class myClass1 {
10:  int a, b;
11:  int getA() {return a;}
12:  int getB() {return b;}
13:  void setAB(int x, int y) {a = x; b = y;}
14: };
15:
16: int main() {
17:  myClass1 a, b;
18:  myStruct1 c;
19:  a.a = 1;
20:  b.b = 2;
21:  c.c = 3;
22:  c.setAB(1, 2);
23:  cout << a.getA() << " " << b.getB() << endl;
24:  cout << c.getC() << endl;
25:  return 0;
26: }
```

Indicate which lines of the above C++ program will result in public/private access violations (detected at compile time).

A. Only lines 11, 12, 13, 19, 20, 21, 22, 23, 24

B. Only lines 6, 11, 12, 13

C. Only lines 19, 20, 21, 22, 23

D. Only lines 11, 12, 13, 19, 20, 21, 22, 23

E. None of the above

Answer: E (in view of a typo in the question, we had erroneously announced C as the answer)

Q5. In this question, we wish to investigate the use of default (D), non-default (N) and copy (C) constructors. Towards this end, consider the following C++ program:

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

class myClass2 { //Class definition
     int a;
   public:
     myClass2() {cout << "D ";}
     myClass2(int x) {cout << "N ";}
     myClass2(myClass2 &x) {cout << "C ";}
 };

 myClass2 myFunc1(myClass2 p) // Function Definition
 {
   myClass2 result = p;
   return result;
 }

 int main() {
   myClass2 v1(35);
   myClass2 v2;
   v2 = v1;
   myClass2 v3 = v2;
   v1 = myFunc1(v3);

   return 0;
 }
```

How many "D", "N" and "C"'s are printed on executing the above program?

A. D:1 N:3 C:0

B. D:1 N:1 C:3

C. D:2 N:1 C:2

D. D:1 N:1 C:2

Answer: B (D:1 N:1 C:3)

Q6. Consider the following C++ program. Assume that all dynamic memory allocations (using 'new') succeed in allocating memory on the heap. What are the values of 'a1' and 'a2' just before the 'main' function returns?

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

int main() {
  int a1 = 10, a2;
  int *p1, *p2, *p3;

  p1 = &a1;
  p2 = new int;
  p3 = p1;
  *p3 = 2 * (*p1);
  *p2 = (*p1) * (*p3);
  p3 = &a2;
  *p3 = 2 * (*p1);

  return 0;
}
```

Select from the following options:

A. Value of 'a1' is 10 and value of 'a2' is 20

B. Value of 'a1' is 20 and value of 'a2' is 20

C. Value of 'a1' is 20 and value of 'a2' is 40

D. Value of 'a1' is 100 and value of 'a2' is 200

E. None of these

Answer: C

Q7. Let 'A' be an array of integers sorted in increasing order. A function for binary search, as studied in class, can be used to quickly find if a specific integer is present in the array 'A'. A student wants to adapt the function for binary search to do the following:

Given an array 'A' of integers sorted in increasing order, and two integers, 'lo' and 'hi', the function should return true if 'A' contains an element 'n' such that lo <= n <= hi. If no such element exists in the array, the function should return false.

The student has written the following incomplete function named 'adaptedBinSrch' to achieve the above purpose. You may assume that the function is called from the 'main' function as 'adaptedBinSrch(A, 0, N, hi, lo)', where 'N' is the size of the array 'A'.

```
// Precondition: A[start] ... A[end-1] is sorted
// in increasing order
// Precondition: lo <= hi
bool adaptedBinSrch(int A[], int start, int end, int hi, int lo)
{
   int first = A[start], last = A[end-1];
   int mid = A[(start +  end)/2];
   if (C1) { return false; }
   if (C2) { return true; }
   if (C3)
     return adaptedBinSrch(A, (start + end)/2, end, hi, lo);
   else if (C4)
     return adaptedBinSrch(A, (start + end)/2, mid, hi, lo);
}
```

You are required to help the student to complete the definition of the function by indicating the right choice of Boolean expressions **C1, C2, C3,** and **C4**. Each of **C1, C2, C3**, and **C4** must be a Boolean expression involving only 'first', 'last', 'mid', 'hi' and 'lo', and nothing else.

Select the ones which are appropriate:

```
A.   C1 is '(last < lo) || (first > hi)'
     C2 is '(first >= lo) && (last <= hi)'
     C3 is 'mid < lo'
     C4 is 'mid > hi'

B.   C1 is '(last > lo) || (first < hi)'
     C2 is '(first <= lo) && (last >= hi)'
     C3 is 'mid > lo'
     C4 is 'mid < hi'

C.   C1 is '(last < lo) || (first > hi)'
     C2 is '(mid >= lo) && (mid <= hi)'
     C3 is 'mid < lo'
     C4 is 'mid > hi'
```

```
D.   C1 is '(last > lo) || (first < hi)'
     C2 is '(mid >= lo) && (mid <= hi)'
     C3 is 'mid < lo'
     C4 is 'mid > hi'
```

E. None of these

Answer: E

Q8. Consider the following C++ program containing a class definition and nested logical blocks in the 'main' function.

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

class myClass {
   int a, b;
 public:
   myClass() {a = 0; b = 1; cout << "+";}
   myClass(int x) {a = x; b = x+1; cout << "-";}
   ~myClass() {  // Destructor Function
      if ((a + b) > 2) {cout << "*";}
      else {cout << "%";}
    }
   void setAB(int x, int y) {a = x; b = y;}
   myClass & operator=(const myClass &z) {
      a = 2*(z.a + z.b); b = 2*a; return *this;
   }
};

int main() {
   myClass v1;
   { // start of block 1
    myClass v2(3);

     { // start of block 2
       myClass v3 = v2;
       v2 = v1;
       v1.setAB(-1, 4);
     } // end of block 2

   } // end of block 1

   return 0;
}
```

What string is printed on executing this program?

A. `+-`

B. `+-***`

C. `+-+++`

D. `+-+**`

E. None of the above

Answer: B

Q9. Consider the following C++ program for reading and writing files. You may assume that 'sizeof(int)' returns 4, i.e. an integer is stored in memory using 4 bytes. You may also assume the following ASCII codes.

Character:	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'
ASCII Code: (in decimal)	48	49	50	51	52	53	54	55	56	57

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

int main() {
  FILE *fp1;
  int a = 0x30303030;       //Note: this is a hexadecimal number

  char c1, c2, c3, c4;

  fp1 = fopen("myFile", "wb");  // Writing a binary file
  if (fp1 == NULL) {return -1;}
  fwrite(&a, sizeof(int), 1, fp1);
  fclose(fp1);

  fp1 = fopen("myFile", "r");  // Reading same file as text
  if (fp1 == NULL) {return -1;}
  fscanf(fp1, "%c%c%c%c", &c1, &c2, &c3, &c4);
  fclose(fp1);

  printf("%c%c%c%c", c1+1, c2+1, c3+1, c4+1);
  return 0;
}
```

What is printed out on executing the above program?

A. 1234

B. 0000

C. 1111

D. 48495051

E. None of the above

Answer: C

Q10. A student has written the following program with her own customed-defined dynamic memory manager.

```
#include<iostream>
using namespace std;
class MemManager {
    long int nBytes;
  public:
    MemManager() {nBytes = 0;}
    int * DynAllocInt(int num) {
      int * result = new int[num];
      if (result != NULL) { nBytes += (num * sizeof(int)); }
      return result;
    }
    void DynDeallocInt(int *p) {
      if (p != NULL) {
          delete p;
          nBytes -= sizeof(int);
      }
   }
    void print() {
      cout << nBytes << " bytes on heap." << endl;
    }
};

int main() {
  int *ptr1;
  MemManager myMem;
  for (int i = 0; i < 10; i++) {
     ptr1 = myMem.DynAllocInt(10);
     if (ptr1 != NULL) myMem.DynDeallocInt(ptr1);
  }
  myMem.print();
  return 0;
}
```

What is printed on executing the above program? You may assume that an integer requires 4 bytes of storage in memory.

A. 400 bytes on heap.

B. 360 bytes on heap.

C. 90 bytes on heap.

D. 40 bytes on heap.

E. None of these

Answer: B or E (any multiple of 36 from 0 to 360 can be printed).

Q11. Consider the following program which manipulates a character array and then prints it as a string.

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

int main() {
 char c[100];

 for (int i = 0; i < 100; i++) {
   c[i] = 'A' + (i % 26);
 }

 for (int i = 0; i < 100; i++) {
   if (c[i] >= 'X') {c[i] = '\0';}
   else if (c[i] <= 'D') {c[i] = '0';}
   else if (c[i-1] == '0') {c[i] = '\0';}
 }

 cout << c;
 return 0;
}
```

What string is printed on executing the above program?

A. 0

B. 0000

C. 0000000000000000000000000

D. None of these

Answer: B

Section 2 – Subjective Questions

Write your answers in 'Section 2 – Answer Sheet' only

Q12. A mathematician wants to write a C++ program for computing the value of a function 'Weird(n)' for values of n >= 0. The mathematician has come up with the following formulae after a long night of hard work.

```
Weird(n) = 2.0 * Weird(n+1) + 3.0 * Weird(n-2) for all n > 2, and
Weird(0) = 1.3, Weird(1) = 2.7 and Weird(2) = 9.1
```

You are required to write a recursive C++ function named 'ComputeWeird' to help the mathematician compute values of Weird(n) for values of n >= 0. Your function MUST use recursion, and must not invoke any function other than 'ComputeWeird' within its body. Also, your function must terminate and produce the correct value of Weird(n) for every value of n >= 0. You may assume that n is an integer. Fill in the incomplete function 'ComputeWeird' shown below.

```
// Precondition: n is an integer >= 0
double ComputeWeird(int n) {
 double result;
```

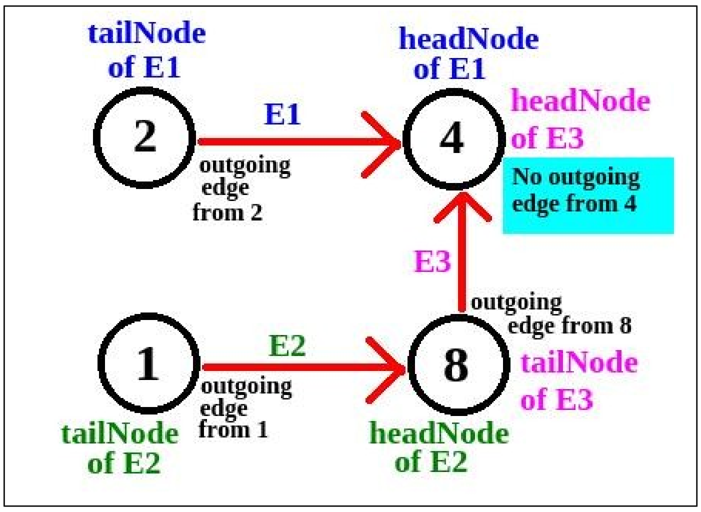
```
 // Write your code in the space allocated for Q12 in the
 //'Section 2 Answer Sheet'
```

```
 return result;
}
// Postcondition: result = Weird(n)
```

Answer:

```
  if (n == 0) result = 1.3;
     else if (n == 1) result = 2.7;
     else if (n == 2) result = 9.1;
     else result = (1/2.0)*(ComputeWeird(n-1) -
                                  3.0*ComputeWeird(n-3));
This can also be solved using switch ... case, or using
conditional expressions.
```

Q13. A directed graph is a set of nodes with directed edges between them. Each node is assumed to have a unique non-negative integer id, and a directed edge is identified by a pair of node ids, called tailNodeID and headNodeID. Specifically, a directed edge is assumed to point from the node with id tailNodeID to the node with id headNodeID.



A student has written the following C++ program to read in a directed graph and print the number of nodes with zero outgoing edges.

```
#include <iostream>
using namespace std;
struct Edge {
  int tailNodeID, headNodeID;
  Edge *next;
  Edge(int tailID, int headID) {
    tailNodeID = tailID; headNodeID = headID; next = NULL;
  }
};
struct Node {
  int id; Edge *outgoing;
  Node *next;
  Node(int ident) {id = ident; outgoing = NULL; next = NULL;}
};
Node *myFunc(Node * &list, int id) {
 Node * result;
 for (result = list; result != NULL; result = result->next) {
       if (result->id == id) break;
 }
```

```
 // Write your code in the space allocated for Q13 in the
 //'Section 2 Answer Sheet'
```

```
 return result;
}
```

```
int main() {
 char response;  int tailID, headID;
 Edge *newEdge;  Node *listOfNodes = NULL;
 Node *tailNode, *headNode;

 // Reading one edge at a time.  Note that
 // no nodes are allocated yet.
 do {
    cout << "Are there more edges? (y/n) ";
    cin >> response;
    if (response == 'y') {
          cout << "Give tail and head node ids of edge: ";
          cin >> tailID >> headID;
          newEdge = new Edge(tailID, headID);
          tailNode = myFunc(listOfNodes, tailID);
          headNode = myFunc(listOfNodes, headID);
          if((newEdge==NULL)||(tailNode==NULL)||(headNode==NULL))
             { return -1; }
          newEdge->next = tailNode->outgoing;
          tailNode->outgoing = newEdge;
    }
    else {break;}
 } while (true);

 // Counting nodes with zero outgoing edges
 Node *temp; int count = 0;
 for (temp = listOfNodes; temp != NULL; temp = temp->next) {
   if (temp->outgoing == NULL) {count++;}
 }
 cout << "Count of nodes with zero outgoing edges: " << count <<
endl;
 return 0;
}
```

Fill in the body of function myFunc to help the student achieve her goal.

Answer:

```
     if (result == NULL) {
       result = new Node(id);
       if (result == NULL) {return NULL;}
       else {result->next = list; list = result;}
     }
```

Inserting the newly allocated node anywhere in "list" is fine. So there could be multiple alternative answers.

Q14. A top-secret message needs to be exchanged between detective agent DetA and his boss DetO through a text file named "DoNotOpenMe". DetA and DetO have agreed that DetA will write the message as a sequence of ASCII characters in file "DoNotOpenMe". Subsequently, when DetO opens the file to read the message, she should only read characters at positions given by the Virahanka number sequence, starting from V(2).

Recall that the Virahanka number sequence is given by: V(0) = 1, V(1) = 1, and V(n) = V(n-1) + V(n-2) for all n >= 2.

Thus, DetO should read the characters at positions V(2), V(3), V(4), ... of file "DoNotOpenMe", until the end-of-file is encountered, in order to reconstruct the secret message that DetA wanted to convey her.

As an example, suppose the file "DoNotOpenMe" contains the sequence of characters "IJHE1L2hL87YUOQS". Since V(2) = 2, V(3) = 3, V(4) = 5, V(5) = 8, V(6) = 13 and V(7) = 21, DetO must read the characters at positions 2, 3, 5, 8 and 13 (the file ends before the 21st character). This gives the sequence of characters "HELLO".

We must help DetO write a program that opens the file "DoNotOpenMe" and prints the decrypted message on the screen.

```
#include <iostream>
#include <cstdio>
using namespace std;
int main() {
 FILE *fp;
 fp = fopen("DoNotOpenMe", "r");
 if (fp == NULL) {
   cout << "Couldn't open DoNotOpenMe" << endl;
   return -1;
 }
 int V_n, V_nMinus1 = 1, V_nMinus2 = 1, n = 1;
 int prevPos = -1; // last position read
 char c1;
 while (true) {
   n++;
   V_n = V_nMinus1 + V_nMinus2;
```

```
 // Write your code in the space allocated for Q14 in the
 //'Section 2 Answer Sheet'
```

```
   fscanf(fp, "%c", &c1);
   if (!feof(fp)) {
     printf("%c", c1);
     prevPos = V_n; // last position read
   }
   else break;
 }
 return 0;
}
```

Fill in the part that should be substituted in the box so that the above program can decrypt DetA's top-secret message. Note that 'feof(fp)' returns a non-zero value if the end-of-file indicator is set for the file pointed to by 'fp'; otherwise it returns zero. The end-of-file indicator is set if a previous read operation attempted to read at or beyond the end of the file pointed to by 'fp'.

Answer:

```
  for (int i = prevPos+1; i < V_n; i++) {
    fscanf(fp, "%c", &c1);
  }
  V_nMinus2 = V_nMinus1;
  V_nMinus1 = V_n;

Variations of this are also possible.
```

Q15. We wish to write a recursive C++ function 'Determinant' that takes an n x n two-dimensional array A of double values as input, treats it as an n x n matrix, and computes the determinant of the matrix as a double value. The computed determinant is communicated to the calling function through one of the parameters (named 'detValue') passed by reference.

The following is a partial definition of the function. If the function succeeds in computing the determinant, we want it to return true. Otherwise, we want it to return false.

```
// Create a copy of the n x n matrix "A" in the (n-1) x (n-1)
// matrix "tempA"
// by excluding the row "excRow" and the column "excCol"
// of matrix "A"
void fillMatrixExcludingRowCol(double **A, double **tempA, int n,
                                              int excRow, int excCol)
{
  int aRow, aCol, tempARow, tempACol;

  for(tempARow=0, aRow=0; tempARow<n-1; tempARow++, aRow++) {
    if (aRow == excRow) {aRow++;}
    for(tempACol=0, aCol=0; tempACol < n-1; tempACol++, aCol++) {
      if (aCol == excCol) { aCol++; }
      tempA[tempARow][tempACol] = A[aRow][aCol];
    }
  }
  return;
}

// A is an n x n two dimensional matrix
bool Determinant(double **A, int n, double &detValue) {

  // Termination case
  if (n == 1) {detValue = A[0][0]; return true;}
  // Allocate memory dynamically for a matrix
  // with dimension reduced by 1
  // Allocating an array of double pointer of size n-1
  double ** tempA = new double*[n-1];
  if (tempA == NULL) {
     cout << "Memory error" << endl; return false;
  }
  for (int i = 0; i < n-1; i++) {
     tempA[i] = new double[n-1];
     if (tempA[i] == NULL) {
        cout << "Memory error" << endl; return false;
     }
  }

  double determinant = 0;
  int multiplier = 1; // either 1 or -1
  for (int i = 0; i < n; i++) {
    // Exclude row 0 and column i from A and copy to tempA
    fillMatrixExcludingRowCol(A, tempA, n, 0, i);
```

```
 // Write your code in the space allocated for Q15 in the
 //'Section 2 Answer Sheet'
```

```
  }

  for (int i=0; i<n-1; i++) delete [] tempA[i];
  delete []  tempA;

  detValue=determinant;
  return true;
}
```

Fill in the part that should be substituted in the box to help achieve the goal of computing the determinant of the matrix A. You should not do any further dynamic memory allocation beyond what is already shown in the partial function implementation above.

Answer:

```
   double result;
   if (!Determinant(tempA, n-1, result)) return false;
   determinant += (multiplier * A[0][i] * result);
   multiplier *= -1;
```

Best Wishes for a Good Score