CS140 Midterm Exam - October 15, 2019

Please answer all questions.
Please put your answers on the answer sheets provided.
Please do not put answers on the exam.

Question 1

Behold the declaration of the **MyClass** class to the right. Please answer the following true/false questions:

```
/*  1 */ class MyClass {
/*  2 */   public:
/*  3 */     MyClass();
/*  4 */     MyClass *Copy();
/*  5 */     MyClass(const MyClass &mc);
/*  6 */     void A(Otherclass &s) const;
/*  7 */     void B(Otherclass s);
/*  8 */     void C(Otherclass &s);
/*  9 */     void D(const Otherclass &s);
/* 10 */     int F;
/* 11 */   protected:
/* 12 */     vector <int> v;
/* 13 */ };
```

A. The copy constructor is declared on line 3.
B. The copy constructor is declared on line 4.
C. The copy constructor is declared on line 5.
D. The method **A()** cannot change **s**.
E. The method **B()** cannot change **s**.
F. The method **C()** cannot change **s**.
G. The method **D()** cannot change **s**.
H. The method **A()** cannot change **v**.

I. The method **B()** cannot change **v**.
J. The method **C()** cannot change **v**.
K. The method **D()** cannot change **v**.
L. When you call **A()**, you will call **Otherclass**' copy constructor.
M. When you call **B()**, you will call **Otherclass**' copy constructor.
N. When you call **A()**, you will call **Otherclass**' regular constructor.
O. When you call **B()**, you will call **Otherclass**' regular constructor.
P. When **A()** returns, it will call the destructor for **s**.
Q. When **B()** returns, it will call the destructor for **s**.
R. This header won't compile, because **F** should be protected.
S. If **x** and **y** are declared as **MyClass** variables, I'm allowed to say "**x=y**", even though I didn't declare an assignment overload.
T. There is a memory leak, because there is no destructor to free up the memory corresponding to the variable **v**.
U. If **Copy()** calls **new**, I should have a destructor that calls **delete**.

Question 2

When the program to the right is executed, what numbers does it print out (circle them on the answer sheet)?

```
int main()
{
  int i;

  i = 0;
  while (i < 10) {
    try {
      cout << i << endl;
      if (i % 2 == 1) throw(i);
      i += 3;
    } catch (int j) {
      cout << i << endl;
      i += j;
    }
    cout << i << endl;
    i += 2;
  }
  cout << i << endl;
  return 0;
}
```

Question 3

A. What is 532 in hexadecimal?
B. Let the binary representation of *x* be 01001010010. What is *x* in hexadecimal?
C. What is 0x48 in decimal?
D. What is (0x48 << 4) in hexadecimal?

Please put your answers on the answer sheets provided.
Please do not put answers on the exam.

Question 4

Below left, I have 17 names, and what each name hashes to with two different hash functions, H1 and H2. The hash values are given in hexadecimal.

Below right, I have a hash table that has been filled in with some of these names.

```
Name                          H1     H2
---------------------      ------ ------
Aidan Nepal                9bc6cc 42757b
Alexander Aeronautic       4345ae 93249d
Austin Prissy              20b849 8c1ef7
Charlie Maiden             0d6fc1 e5855d
Claire Tva                 fe30bf 8124af
Gavin Parallelepiped       0e9b65 db8914
Hunter Fabric              b112a2 78d98d
Isabelle Stack             4c30e6 f7193f
Joshua Polonium            831e15 eeba38
Madison Willoughby         1874dc 911d61
Max Mere                   fc7e5b 8d1814
Maya Paddy                 107b1e 5e84ff
Natalie Sober                   ?      ?
Noah Porous                ae3e75 2dbc99
Riley Predecessor          f001c1 385e34
Savannah Tradesman         a3d7ee ab7176
Sophia Keys                69147a 493120
```

```
0.  --------------------
1.  Charlie Maiden
2.  Hunter Fabric
3.  --------------------
4.  --------------------
5.  Gavin Parallelepiped
6.  Isabella Stack
7.  --------------------
8.  --------------------
9.  --------------------
10. Sophia Keys
11. Max Mere
12. Aiden Nepal
13. Madison Willoughby
14. Savannah Tradesman
15. Claire Tva
```

Please answer the questions below. Do not answer the questions as if one affects the other. Answer them all with respect to the tables above. For example, you should not answer part *C* as if Austin Prissy were inserted into the table. Instead, you simply answer with respect to the table above.

In all of the questions, assume that hash function H1 is used to insert into the table, and if double-hashing is used, then hash function H2 is used as the second hash function.

A. What is the load factor of the table (you can give a fraction here)?
B. Into which index will Austin Prissy go into the table, using linear probing?
C. Into which index will Joshua Polonium go into the table, using quadratic probing?
D. Into which index will Maya Paddy go into the table, using double hashing?
E. Into which index will Noah Porous go into the table, using linear probing?
F. Into which index will Alexander Aeronautic go into the table, using quadratic probing?
G. Into which index will Riley Predecessor go into the table, using double hashing?
H. When using double hashing, Natalie Sober cannot be entered into the table. Give me values of H1 ahd H2 that will make this happen.
I. How do we know that Madison Willoughby was inserted into the table after Aiden Nepal?

Please put your answers on the answer sheets provided.
Please do not put answers on the exam.

On Questions 5 through 8:

- Please do not bother with "#include" statements, or with "using namespace std." I don't want you wasting your time with that.
- You should uses **const** and reference variables correctly.
- I will deduct if you intermix code and variable declarations.
- I will also deduct if your implementation is overly wasteful in terms of time or space.
- Vectors, lists and deques have the following methods: **push_back()**, **pop_back()**, **size()**, **insert()**, **erase()**.
- Lists and deques also have: **push_front()**, **pop_front()**.

Question 5

Write a program that prints (on standard output) its first command line argument that is an integer, and a newline, and then exits. If there is no command line argument that is an integer, it should print "No integer" and a newline, on standard error.

```
UNIX> ./a.out One 2 Buckle my Shoe
2
UNIX> ./a.out 1 2 Buckle my Shoe
1
UNIX> ./a.out Fred > /dev/null
No integer
UNIX>
```

```
UNIX> ./a.out
No integer
UNIX> echo 1 2 3 | ./a.out
No integer
UNIX> echo 1 2 3 | ./a.out I break 4 food.
4
UNIX>
```

Question 6

Write a procedure called **vreverse()**, which takes as input a reference parameter that is a vector of integers. It should reverse the elements of the vector. In other words, if the input vector is `{ 1, 20, 2, 10, 3 }`, then when **vreverse()** is done, the vector should be `{ 3, 10, 2, 20, 1 }`.

Question 7

Write a procedure called **dmiddle()**, which takes as input a pointer to a list of strings, and returns an integer. It should do the following:

- If the list is empty, do nothing.
- If the list has an odd number of elements, it should delete the middle element.
- If the list has an even number of elements, it should delete the two middle elements.
- It should return the number of elements that it deleted.

Please put your answers on the answer sheet.

Question 8

Below are class specifications for two classes: **Person** and **Data**. The **Data** class has three methods -- a constructor, a destructor, and **Print_By_Decade()**. It stores people in two data strutures -- a vector that has all people, and a vector of vectors called **Decades**. Each vector in **Decades[i]** is a list of people who are between *10i* and *10i+9* years old.

The method **Print_By_Decade()** should take the value *i* and do the following. If **Decades[i]** does not exist, then it should return false. Otherwise, it should print the names of all people who are between *10i* and *10i+9* years old. It should print them one per line and return true.

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

/*  1 */   class Person {
/*  2 */     public:
/*  3 */       string name;
/*  4 */       int age;
/*  5 */   };
/*  6 */
/*  7 */   class Data {
/*  8 */     public:
/*  9 */       Data();
/* 10 */       ~Data();
/* 11 */       bool Print_By_Decade(int decade);
/* 12 */     protected:
/* 13 */       vector <Person *> All;
/* 14 */       vector < vector <Person *> > Decades;
/* 15 */   };
/* 16 */
/* 17 */   Data::Data()
/* 18 */   {
/* 19 */     string n;
/* 20 */     int a;
/* 21 */     int decade;
/* 22 */     Person *p;
/* 23 */
/* 24 */     while (cin >> n >> a) {
/* 25 */       decade = a/10;
/* 26 */       p = new Person;
/* 27 */       p->name = n;
/* 28 */       p->age = a;
/* 29 */       if (decade >= Decades.size()) Decades.resize(decade+1);
/* 30 */       All.push_back(p);
/* 31 */       Decades[decade].push_back(p);
/* 32 */       printf("%s %d\n", n.c_str(), (int) Decades[decade].size());
/* 33 */     }
/* 34 */   }
```

You have four jobs:

A. Implement **Print_By_Decade**().
B. Implement the destructor.
C. I haven't put any **const** declarations here. If you see where to put one or more of them, then do so (say something like, "on line 20, turn 'int a' into 'const int a'").
D. Suppose I wanted to error check in the constructor, that the value of **a** is non-negative. Write the code to do that (say something like, "before line 27, insert the following code...")