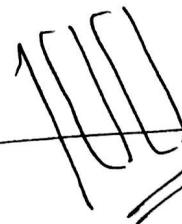


NAME: Khaja Anwar Ali Siddiqui

CSCI 589-2



Fall 2015

Midterm Exam 1
Closed Book & Notes
(100 points)

- ✓ 1. What code in C++ could you execute to figure out the character whose ASCII value is 100? Your code needs to print out the unknown character on stdout.

```
#include <iostream>
using namespace std;
int main()
{
    int i = 100; char c = (char)i;
    cout << "The character at " << i << " is " << c << endl;
    return 0;
}
```

2. For any given values of a , b , and c , is there anything wrong with the following C++ code?

```
int x = a, y = b, z = c;  
if (x < y < z) cout << x << " < " << y << " < " << z << endl;
```

Eg $x=1 ; y=2 ; z=3.$

If $\begin{array}{c} (1 < 2 & 3) \\ \downarrow \\ 1, 0 \end{array}$

Ans:- The relational operators are evaluated from Left to Right.

So, $x < y$ can either evaluate to 0 (if $x > y$) or 1 (if $x < y$).

So, if $z \geq 1$, the condition evaluates true, irrespective of $x & y$ values.

\Rightarrow If condition doesn't allow such declarations.

3. Write a C++ program that reads in a temperature from the stdin in degrees Celsius (C) and displays the corresponding temperature on stdout in degrees in Fahrenheit (F). The conversion formula is: $F = (9/5)C + 32$.

```
#include <iostream>
using namespace std;
int main(){
    double C, f;
    cout << "Provide temperature in celsius: " << endl;
    cin >> C;
    f = ((9.0 * C) / 5.0) + 32;
    cout << "Temperature in Fahrenheit: " << f << endl;
    return 0;
}
```

4. Write a C++ program to add integers between a and b (inclusive), where $a \leq b$.

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

int main()
{
    int a, b;
    int sum = 0;

    cout << "Provide two numbers to calculate the
          sum of integers in between" << endl;
    cout << "First Number (from): " << endl;
    cin >> a;
    cout << "Second Number (till): " << endl;
    cin >> b;
    if (a > b) cout << "Wrong input!" << endl;
    if (a < b) {
        for (int i=a; i<=b; i++)
            sum += i;
    }
    if (a==b) sum = 0;

    cout << "The sum is " << sum << endl;
    return 0;
}
```

5. What is the purpose of the following C++ function?

```
1 unsigned F ( unsigned n ) → suppose n = 1234
2 {
3     unsigned s = 0; // 1234 > 0
4     while ( n > 0 ) { // 1234 % 10; 1234 / 10;
5         s += n % 10; // S = 4 (can be from 0-9).
6         n /= 10; // infinite loop. S+ = 4; S = 0 (always).
7     }
8     return s;
9 }
```

Ans:- The above program generates an infinite loop.
Since n is declared as unsigned, provided that some
of the compiler treats 'unsigned' as "unsigned int".

6. Write a C++ program segment for the followings:
- Create an empty C++ string.
 - Assign the name "Michael J. Fox" to the string.
 - Erase the middle initial of the name in the string.
 - Replace the first name "Michael" with "Mike" in the string.
 - Create a new string from the existing one as the name in the new string will be "Fox, Mike".

(a) ~~std::string str~~ ~~✓~~ insert(str, n)

(b) ~~str = "Michael J. Fox";~~ ~~✓~~ erase(pos, n)

(c) ~~str = str.erase(7, 1); // Michael Fox~~ ~~✓~~ replace(str, pos, n)

(d) ~~str = str.replace("Mike", 0, 7); // Mike, Fox~~ ~~✓~~ replace(str, pos, n)

(e) string str1, str2;
str1 = substr(str, 5); // for Fox
str2 = substr(str, 0, 5); // for Mike
str = str1 + (string)", " + str2;

- ~~7.~~ 2 For the Point class whose partial definition is given below, show the implementations of the methods setX() and get X().

```
class Point {  
    int x, y;  
public:  
    void setX ( const int& );  
    int getX ( ) const;           // assign a value to x  
};                                // return value of x
```

Ans
 Point :: void setX (const int& u)

{

x = u;

Point :: int getX () const;

{

return x;

}

8. What is the output of the following C++ program?

```
→ string intToString ( const int& );  
→ class Date {  
    friend ostream& operator<< ( ostream&, const Date& );  
public:  
    Date ( const string& s ) { stringToDate ( s ); } // 11-11-1918  
    void setDate ( const string& s ) { stringToDate ( s ); } // 14/8/1945  
private:  
    int day, month, year;  
};  
  
→ void Date :: stringToDate ( const string& s )  
{  
    char a, b;  
    istringstream is ( s ); // 14/8/1945  
    is >> day >> a >> month >> b >> year; //  
}  
    14 / 8 / 1945  
→ ostream& operator<< ( ostream& os, const Date& d )  
{  
    os << intToString ( d.month ) << " " << d.day << ", " << d.year;  
    return os;  
}           November 11, 1918  
  
→ int main ()  
{  
    Date peace ( "11-11-1918" ); //  
    cout << "World War I ended on " << peace << ".\n"; // 1  
  
    peace.setDate ( "14/8/1945" ); //  
    cout << "World War II ended on " << peace << ".\n";  
    return 0;  
}
```

```
→ string intToString ( const int& n )  
{  
    switch ( n ) {  
        case 1: return "January";   case 2: return "February";  
        case 3: return "March";     case 4: return "April";  
        case 5: return "May";       case 6: return "June";  
        case 7: return "July";      case 8: return "August";  
        case 9: return "September"; case 10: return "October";  
        case 11: return "November"; case 12: return "December";  
        default: return "";  
    }  
}
```

Ans O/P

World War I ended on November 11, 1918.
World War II ended on August 14, 1945.

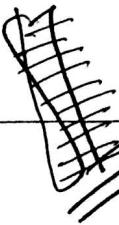
-
+4
-13
89

NAME: Khaja Anwar Ali Siddiqui

CSCI 589-2

Midterm Exam 2
Closed Book & Notes
(100 points)

Fall 2015



1. An empty vector container is declared as vector < int > intList. Write C++ statements for the followings:
- Add integers 13, 75, 28, and 35 in intList.
 - Double the value of each element in intList.
 - Print out the contents of intList, on stdout on a single line, separated by spaces.

Ans:-

`vector<int> intList;`

a) `intList.push_back(13);
intList.push_back(75);
intList.push_back(28);
intList.push_back(35);`

b) `vector<int>::iterator it; int temp;`

`for (it = intList.begin(); it != intList.end(); it++) {
 temp = *it; // stores "13".
 intList.erase(it); X-2
 intList.push_back(2 * temp);
}`

c) `{
 cout << *it << " ";
}`

`13 | 75 | 28 | 35
↓
13
↓
28 35 26
↓
28
↓
28 35 26 150
↓
35 26 150 56.
↓
35
↓
26 150 56 70`

~~X~~

2. What is the output of the following C++ program?

```

const unsigned N = 5;

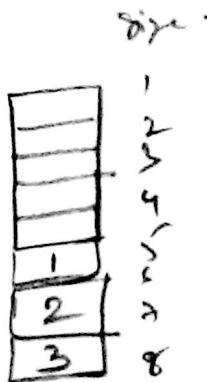
int main ()
{
    vector< int > v; ✓
    cout << v.size () << endl; // 0

    v.resize ( N ); cout << v.size () << endl; // 5
    v.push_back ( 1 ); v.push_back ( 2 ); v.push_back ( 3 );
    cout << v.size () << endl; // 8
    for ( unsigned i = 0; i < v.size (); i++ ) cout << v [ i ] << ' ';
    cout << endl; // 0 0 0 0 0 1 2 3

    v.pop_back (); cout << v.size () << endl;
    cout << v.front () << '\t' << v.back () << endl;

    return 0;
}

```



O/P :-

0 /
5 /
8 /
0 0 0 0 0 1 2 3 ← *space* ← *c* -

7
0 0 0 0 0 1 2 3

~~3~~

3. Consider the following recursive function:

```

void recFunc ( int x )
{
    if ( x >= 10 ) {
        recFunc ( x / 10 );
        cout << x % 10 << endl;
    }
    else cout << x << endl;
}

```

What is the output of the following statements? (a) recFunc (258); (b) recFunc (7);
(c) recFunc (36); (d) recFunc (-85);

- a) 852
- b) 71
- c) 63
- d) -85

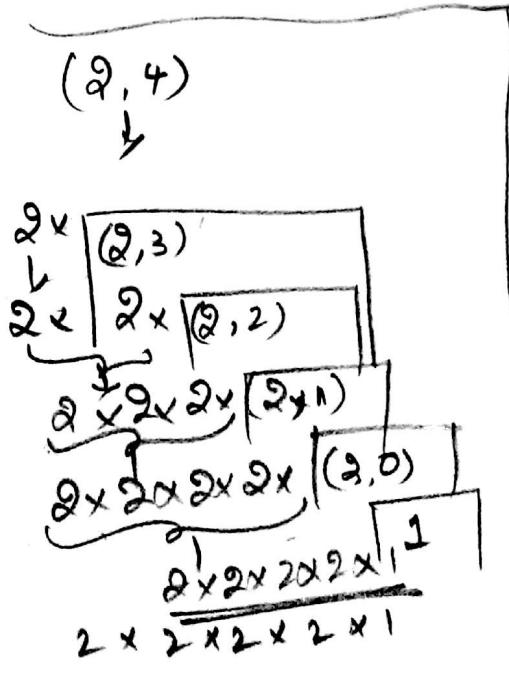
258
↓
recF(25).

↓
8
↓
recF(2).
↓
5
↓
2.

recF(-85)
↓
-85.

- ~~Q~~ 0.5
 * Write a recursive C++ function: `unsigned raiseToPower (unsigned n, unsigned k)` that calculates n^k . The recursive insight that you need to solve this problem is the mathematical property that $n^k = 1$ if k is 0; otherwise it's equal to $n * n^{k-1}$.

```
Unsigned raiseToPower(unsigned n, unsigned k) {
    if (k == 0) return 1;
    else return n * raiseToPower(n, k-1);
}
```



$$\begin{aligned}
 n^k &= (2^3 \times 2^1)^4 \\
 &= 2^3 \times 2^1 \times 2^1 \times 2^1 \\
 &= 2^3 \times 2^1 \times 2^1 \times 2^1 \\
 &= 2^0 \times 2^1 \times 2^1 \times 2^1 \\
 &= 1 \times 2^1 \times 2^1 \times 2^1
 \end{aligned}$$

+²

5. Consider the definition of the following class:

```
class testClass {  
public:  
    int sum (); // returns the sum of private data members  
    void print () const; // prints out values of private data members  
    testClass (); // default constructor - initializes private data  
    // members to 0  
    testClass ( int a, int b ); // constructor - initializes private data members to  
    // values a and b, respectively  
private:  
    int x, y;  
};
```

- (a) Write the definition of the member functions as described in the definition of the testClass.
(b) Write a prototype of a new constructor of the testClass that combines the prototypes of its existing two constructors into a single one by using default values.

(a)

```
int testClass::sum () {  
    return (x+y);  
}  
  
void testClass::print () const {  
    cout << "x is : " << x << "y is : " << y << endl;  
}  
  
testClass::testClass () {  
    x=y=0;  
}  
  
testClass::testClass ( int a, int b ) {  
    x=a; y=b;  
}  
  
public  
testClass::testClass ( int a=1; int b=2 );  
// where 1,2 are assumed to be default values?
```

X-12.5

6. What is the output of the following C++ program?

```
class X {  
public:  
    X( int = 0 );  
    void print() const;  
private:  
    int sz;  
    int* p;  
};
```

```
X::X( int n )  
{  
    sz = n; p = 0;  
    if ( sz > 0 ) p = new int[ sz ];  
    for ( int i = 0; i < sz; i++ ) p[ i ] = i;  
}
```

```
void X::print() const  
{  
    for ( int i = 0; i < sz; i++ ) // OK  
        cout << p[ i ] << ' ';  
    cout << endl;  
}
```

```
int main()  
{  
    const int N = 5;  
    X a, b( N ), c = b; // a, b, c  
    cout << "a: "; a.print();  
    cout << "b: "; b.print();  
    cout << "c: "; c.print();  
    cout << endl;  
  
    return 0;  
}
```

Ans: ~~OK~~

~~Compilation Error~~

a:

b: 0 1 2 3 4

c: 0 1 2 3 4

6

0 1 2 3 4

sz = 5; p = 0;
p = new int[5];

p[0] = 0;

1

2

3

4

- ~~7~~²
7. An empty set container is declared as set < string > s. Write C++ statements for the followings:
- Get strings from stdin and put them in s until EOF.
 - Print out the strings, which start with a capital letter, in s on stdout, on a separate line for each string.

Set<string> s;

a)

```
string str;
while (getline(cin, str)) {
    s.insert(str);
}
```

b)

```
Set<string>::iterator i;
for (i = s.begin(); i != s.end(); i++)
{
    string str;
    str = *i;
    char c = str[0];
    if ('C' >= 'A' && C <= 'Z')
        cout << str << endl;
}
```

8. Write a C++ program that (a) declares an array alpha of 50 components of type double; (b) initialize the array so that the first 25 components are equal to the square of the index variable; (c) and the last 25 components are equal to three times the index variable; (d) output the array so that 10 elements per line are printed.

a)

```
double alpha[50];
```

b)

```
for (int i=0 ; i<25 ; i++)  
    alpha[i] = i*i;
```

c)

```
for (int i=25 ; i<50 ; i++)  
    alpha[i] = 3*i;
```

d)

```
for (int i=0 ; i<50 ; i++)  
{  
    cout << alpha[i];  
    if (i%10 == 0) cout << "\n";  
}
```

NAME: Khaja Anwar Ali Siddiqui

6.75

CSCI 689-2

Quiz 1

Closed Book & Notes
(10 points)

Fall 2015

1. Answer each part of this question.

- When you write a C++ program, do you prepare a source file or an object file?
- What characters are used to mark comments in a C++ program?
- In a #include line, the name of the library header file can be enclosed in either angle brackets or double quotation marks. What is the difference between the two forms of punctuation?
- How would you define a constant called CENTIMETERS_PER_INCH with the value 2.54?
- What is the name of the function that must be defined in every C++ program? What statement typically appears at the end of that function?
- What is the purpose endl when you are writing to the cout output stream?
- How do the types short, int, and long differ?
- List all possible values of type bool.
- What statements would you include in a C++ program to read a value from the user and store it in the variable x, which is declared as a double?
- Indicate the values and types of the following expressions:

Q. 25

x i.	$2 + 3 = 5$	int
x ii.	$19 / 5 = 3$	int
x iii.	$19.0 / 5 = 3.8$	double
x iv.	$19 \% 5 = 4$	int
x v.	$2 \% 7 = 2$	int

Ans 1 (a) A source is created when we write a C++ program.

(b) C++ supports two types of comments.
 i) Single-line using // (double forward slashes)
 ii) Multi-line comment using /* and */

(c) When name of header file is enclosed in double quotation marks it is a user-defined header file, if its enclosed in angular brackets it is a pre-defined header file.

- const float CENTIMETERS_PER_INCH = 2.54;
- main() must be declared in every C++ program, it ends with 'return 0';
- 'endl' is a stream manipulator, which defines the end of current line in cout stream.
- short, int & long differ in their range. short, int can be of same range but long must be greater than int & short.
- true, false
- double x;
- Cin >> x;

-0.25

2. Write a C++ program that reads in a list of integers from the user until the user enters the value 0 as a sentinel. When the sentinel appears, your program should display the largest value in the list. (a) Be sure to define the sentinel value as a constant in a way that makes it easy to change; (b) You should also make sure that the program works correctly if all the input values are negative; and (c) You should also not use a break statement in the program.

```
#include <iostream>
using namespace std;
const int SENT=0;
```

```
int main() {
```

```
    int x, y, max;
```

```
    y = 0;
    cout << "List the numbers whose max is to be calculated '0' to terminate:";
    cout << endl;
    cin >> x;
    while(true) {
        max = y;
        if(x > y) max = x;
        if(x == SENT) break;
        cout << "Input x: ";
        if(x > y) max = x;
        cout << endl;
        cout << "The maximum is " << max << endl;
    }
}
```

Annotations:

- "y = 0;" → "Can't assume this b/c user can enter negative #'s."
- "cout << "List the numbers whose max is to be calculated '0' to terminate:";" → "Can't assume this b/c user can enter negative #'s."
- "cout << endl;" → "Should make while (x != SENT)"
- "if(x > y) max = x;" → "Don't use break!"
- "if(x == SENT) break;" → "Should make while (x != SENT)"
- "cout << "The maximum is " << max << endl;" → "Compare to max." "Y = 0 every time."

Instead
 $\text{cin} \gg x$ once
 before while loop.
 & set $\text{max} = x$.

-3

NAME: Answer

(8)

CSCI 689-2 Please write your full name.

Quiz 2
Closed Book & Notes
(10 points)

Fall 2015

1. Answer each part of this question.

- a) What is the difference between a *character* and a *string*?
- b) True or false: If you execute the lines
`string line; cin >> line;` *S skips by whitespace*
the program will read an entire line of data from the user and store it in the variable line.
- c) Which arguments to the `getline` function are passed by reference?
- d) True or false: In C++, you can determine the length of the string stored in the variable str by calling `length (str)`.
- e) If you call `s1.replace (0, 1, s2)`, which is the *receiver*?
- f) What is the effect of the + operator when it is used with two string operands?
- g) When you select an individual character from a C++ string, you can use either the at method or the standard subscript notation in which the index is enclosed in square brackets. From the client's perspective, what is the difference between these two options?
- h) When C++ evaluates the expression `s1 < s2`, what rule does the string class use to compare the string values?
- i) What value does the `find` method return to indicate that the search string does not appear?
- j) How can you convert a primitive string value to a C++ string? How can you specify a conversion in the opposite direction?

- a) A character occupies one byte of memory, is capable of storing / displaying a single character from the set of ASCII character list. A string is collection of characters and occupies memory based on its length.
- b) False.
- c) `cin, x.`
- d) False. (Syntax should be `str.length();`)
- e) `s1` is the receiver.
- f) + operator concatenates two strings.
- g) at method option ensures the presence of valid character or `npos`. where indexing might return a garbage value if character is not found.
- h) The outcome of `s1 < s2` depends on 'Lexicography' i.e. based on order of dictionary.
- i) `std::string`.
- ~~A primitive string can be converted to C++ string by `string str(string primitiveStr);`~~
- ~~To reverse C++ string to primitive `string.str()` should be used.~~

T-0.5

2. Write a C++ function capitalize (str) that returns a string in which the initial character is capitalized (if it is a letter) and all other letters are converted to lowercase. Characters other than letters are not affected. For example, both capitalize ("BOOLEAN") and capitalize ("boolean") should return the string "Boolean".

String
↓
string capitalize (str) {

```
int l = str.length();
int factor = 'A' - 'a'; // conversion factor from ASCII table
char first_ch = str[0];
if (first_ch > 'a') str[0] += factor; cout << str[0];
for (int i = 1; i < l; i++) {
    if (str[i] >= 'A' && str[i] <= 'Z') str[i] -= factor;
    cout << str[i];
}
}
```

\uparrow needs to return a string. Not cout

Function declaration ~~return value~~: O/1

to Upper : 1/1

to Lower : 1/1

return ~~value~~ : ~~return~~ O/1

str.length() : 1/1

Total = 3.5/5

(-1.5)

NAME: Khaja Anwar Ali, Siddiqui.

8

CSCI 689-2

Quiz 3
Closed Book & Notes
(10 points)

Fall 2015

1. Answer each part of this question.

- a) What are the three standard file streams defined by the <iostream> library?
- b) What are the formal names for the << and >> operators?
- c) What value do the << and >> operators return?
- d) What is a *manipulator*?
- e) What is the difference between a *transient* and a *persistent* property?
- f) Describe how the fixed and scientific manipulators change the format for floating-point output? What happens if you don't specify either of these options?
- g) What is the purpose of the types ifstream and ofstream?
- h) The argument to open must be a C-style string? How does this requirement affect the code you write to open a file?
- i) How can you determine if an open operation on a stream was successful?
- j) When you are using the getline method to read a file line by line, how do you detect the end of the file?

- a) cin, cout, cerr
- b) << is called insertion operator & >> is called extraction operator.
- c) << returns the value to the left side of the standard console & >> reads the value from right side to left side.
- d) A manipulator performs output formatting like width, precision etc.
- e) A transient property lasts for current usage. A persistent property lasts unless modified explicitly.
- f) fixed manipulators would display output in numbers (decimal System). Scientific manipulators would display in scientific notation eg. 0.3145e+01 If not specified, the output would be in decimal form with rounded off values.
- g) ifstream & ofstream declares variables for reading and writing to files respectively.
- h) While coding file name must be passed in C-style string. Else open method should be appended as infile.open(string, C-STR());
- i) If (!infile.open("filename")) cout << "open successful"; failed.
- j) while(infile); i.e infile remains true till last line.

2. What is the output of the following C++ program?

```
const double PI = 3.14159265358979323846;
void printPI();
```

```
int main()
{
    printPI();
    cout << fixed; printPI();
    cout << scientific; printPI();
    return 0;
}

void printPI()
{
    for (int p = 0; p <= 6; p += 2) {
        cout << setw(2) << p;
        cout << setw(14) << setprecision(p) << PI << endl;
    }
    cout << endl;
}
```

P=0	B2	R4	P.G.
0-	2-	4-	6-
0			

$\text{setprecision is } \# \text{ after decimal.}$

Output: i.e. $\text{setprecision}(2) = 3.1\cancel{4} \times 10^0$

0 3

2 3.1

4 3.14

6 3.1416

0 3.141592

3.141592

fixed

0 3

2 3.1

4 3.14

6 3.1416

0 3.141592

Scientific

0+e00 3+e00

2+e00 3.1+e00

4+e00 3.141+e00

6+e00 3.141592+e00

9

NAME: Khefa Anwar Ali Siddique (Z1778291)

CSCI 689-2

Quiz 4
Closed Book & Notes
(10 points)

Fall 2015

1. Answer each part of this question.

- When you are using the get method to read a file character by character, how do you detect the end of a file?
- Why is the return type of get declared as int instead of char?
- What is the purpose of the unget method?
- What is the purpose of the types istream and ostringstream?
- True or false: ws manipulator can be used to delete whitespaces in an input stream?
- Define each of the following terms: class, object, method.
- In a C++ class definition, what do the keywords public and private mean?
- True or false: In C++, the only difference between the keyword struct and the keyword class is that struct makes fields public by default.
- What operator does C++ use to select an instance variable from an object?
- What are getters and setters?

~~You didn't use get...~~

~~if ((c = getch(a, x)) != EOF) returns false; It means end of file is reached.~~

~~Since, the end of file is outside the character set of each machine might have diff' definitions for EOF, so get must return an int value. EOF = -1.~~

- unget method places the stream pointer back to last character read, so that the last character is read again for next execution.
- istream & ostringstream are used to define variables for reading & writing strings from a file. They declare variables for manipulating strings.
- False.
- A class is a group of variables & methods, encapsulated ~~for~~ within a definition. An object is an instance of a class. A method is code defined inside a class.
- 'Public' & 'private' are access modifiers. By 'public' definition the scope is defined outside the block. 'Private' declares an object/class variable with limited scope or within current block.
- False (class can have methods unlike structures).
- Reference operator ". " return values to client. Getters takes the ~~values~~ (or) give values to client. Setters allows clients to set values to current execution.
- Getters takes the ~~values~~ (or) give values to client. Values to current execution.

-1.0

2. What is the output of the following C++ program?

```
class Point {  
public:  
    void setX ( string xx ) { x = xx; }  
    void setY ( string yy ) { y = yy; }  
    string getX ( ) const { return x; }  
    string getY ( ) const { return y; }  
    void print ( ) {  
        cout << "(" << convert ( x ) << "," << convert ( y ) << ")\n";  
    }  
private:  
    string double x, y;  
    double convert ( string s ) {  
        double d;  
        istringstream stream ( s ); stream >> d;  
        return d;  
    }  
};  
  
int main ( )  
{  
    Point p; // creates obj: p  
    string u = " 7.3x ", v = " 11.5y ";  
    p.setX ( u ); p.setY ( v ); // passes u, v  
    cout << p.getX ( ) << endl; // 7.3x  
    cout << p.getY ( ) << endl; // 11.5y  
    p.print (); //  
  
    ← string w1 = u + v, w2 = v + u;  
    p.setX ( w1 ); p.setY ( w2 );  
    cout << p.getX ( ) << endl;  
    cout << p.getY ( ) << endl;  
    p.print ();  
  
    return 0;  
}
```

Output →

7.3x

11.5y

(7.3, 11.5)

7.3x 11.5y

11.5y 7.3x

(7.3, 11.5)

5/5

NAME: Khaja Anwar Ali Siddiqui

(95)

CSCI 689-2

Quiz 5
Closed Book & Notes
(10 points)

Fall 2015

1. Answer each part of this question.

- a) What is the STL?
- b) List at least three advantages of the vector class over the more primitive array structures available in C++.
- c) What type name would you use to store a vector of Boolean values?
- d) True or false: The default constructor for the vector class creates a vector with 10 elements, although you can make it longer.
- e) How would you initialize a vector `< int >` with 20 elements, all equal to 0?
- f) What method do you call to determine the number of elements in a vector?
- g) Why is it important to pass vectors and other collection objects by reference?
- h) What is the difference between the *copy constructor* and the *assignment operator* for the vector class?
- i) Write a C++ statement to increase the size of the vector `v` by 10.
- j) Write C++ statement to check if a vector is empty, and if it's not, make it empty.

2. Standard template Library, that defines templates for various parameterized type classes (or containers).

b) i) Unlike vectors, arrays have fixed size.

ii) Arrays need an additional iterator to keep track of index.

iii) Unlike arrays, Vectors allow insertion and deletion at a particular index.

3) `vector<bool> name;` // declares vector for boolean values.

d) False.

e) `vector<int> v(20);` // Has all elements zero.

f) `vector.name.size();`

8) Call by reference makes changes to container without making a copy for which changes are reflected in calling function too. If a copy is made and not returned, the changes may lost.

Copy constructor copies values including elements storage. Unlike assignment operator, which has to deal with individual elements.

9) `v.resize(v.size() + 10);` (i) If (!v.empty()) `v = v.clear();`

- 0.5

2. In statistics, a collection of data values is often referred to as a *distribution*. The most common statistical measure is the *mean*, which is simply the traditional average. Write a C++ function: double mean (const vector<double>& data); that returns the mean of the data in the vector.

double mean(const vector<double>& data)

{

 double Vmean = 0, Vsum = 0;

 int Vsize = data.size();

 for(int i=0; i < Vsize; i++)

 Vsum += data[i];

 Vmean = (Vsum/Vsize);

 return Vmean;

- The first ball triggers two balls from mousetrap.
 → Either they can land on floor
 → or land on mousetrap ↗
 → Once a ball falls on a mousetrap,
 ball stops & takes no further action
 ↓
 should be
 selected randomly
 ↗
 → Balls bounce in room & land
 on floor after few cycles
 (min = 1, max = 4).
 ↓
 if already
 sprung.
 ↗
 sprung no
 else
 two
 more
 ball

- 25x25 grids.
 → Each grid has two balls.
 → At first an additional ball is released.

- Simulation runs until
 no more balls in air.

NAME: Khaja Ammar Ali Siddiqui (92)

CSCI 689-2

Quiz 6
Closed Book & Notes
(10 points)

Fall 2015

1. Consider the following recursive function:

```
int mystery ( int n )
{
    if ( n == 0 ) return n; //trivial case
    else return ( n + mystery ( n - 1 ) );
}
```

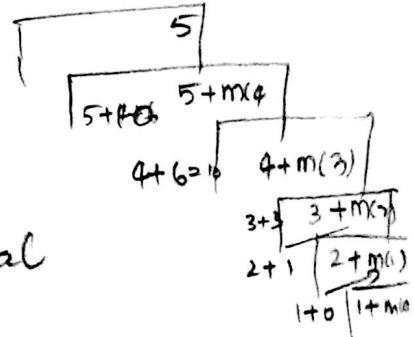
- a) What valid values can be passed as parameters to the function mystery ()?
- b) If mystery (0) is a valid call, what is its value? If not, explain why.
- c) If mystery (5) is a valid call, what is its value? If not, explain why.
- d) If mystery (-3) is a valid call, what is its value? If not, explain why.

Q2 Valid values would be from 0 to positive values of n
i.e (0, n)

b) mystery (0) is a valid call. Since the loop returns as soon as
if ($n==0$) returns true Output would be '0'.

c) mystery (5) is a valid call, value would be '15'.

d) mystery (-3) is an invalid call, since it calls (-4)
which inturn calls mystery (-5) & so on. The trivial
case of $n=0$ will not be reached.



2. Spherical objects, such as cannonballs, can be stacked to form a pyramid with one cannonball at the top, sitting on top of a square composed of four cannonballs, sitting on top of a square composed of nine cannonballs, and so forth. Write a recursive C++ function `cannonball()` that takes as its argument the height of the pyramid and returns the number of cannonballs it contains. Your function must operate recursively and must not use any iterative constructs, such as while or for.

Level 1: 1 → 3
 Level 2: 9 → 5
 Level 3: 9 → 3

```
int Cannonballs(int n) {
    just pick one to
    be the base case. . .
    if (n == 0 || n == 1) return n;
    else return n + Cannonballs(n - 1);
}
```

can't code n^2 , write $n \times n!$

4.7.5

2 - 0.3

NAME: Khoya Anwar Ali Siddiqui

CSCI 689-2

Quiz 7
Closed Book & Notes
(10 points)

Fall 2015

- 1) What is the output of the following C++ program segment?

```
int main ()  
{  
    const int N = 10; int x = 1; int a [ N ]; // a[10]  
    int* p = a;  
  
    a [ 0 ] = x; *p = 5;  
    cout << "a [ 0 ] = " << a [ 0 ] << endl; // 5  
  
    p = &a [ 1 ]; *p = 10; // a[1] = 10,  
    cout << "a [ 1 ] = " << a [ 1 ] << endl; // 10  
    cout << "* ( a + 1 ) = " << *( a + 1 ) << endl; // 10  
    cout << "* ( &a [ 1 ] ) = " << * ( &a [ 1 ] ) << endl; // 10  
  
    int* q = a + N - 1; *q = 50; // 1000 + 9 * 50 = a[10] = 50  
    cout << "a [ N - 1 ] = " << a [ N - 1 ] << endl; // 50  
    return 0;  
}
```

$a = 1000$

1	X/5	P
2	10	
3		
4		
5		
6		
7		
8		
9		
10		

$1000 + 9 \times a$

Q1

$$a[0] = 5$$

$$a[1] = 10$$

$$*(a+1) = 10$$

$$*(\&a[1]) = 10$$

$$a[N-1] = 50$$

- 2) Write a recursive C++ function with prototype: void decToHex (unsigned num); to convert a given nonnegative integer num in a decimal format—that is, base 10—into the equivalent hexadecimal number—that is, base 16—and print out the resulting hexadecimal number on stdout. Your function must operate recursively and must not use any iterative constructs, such as while or for. Hint: Use the following array to represent the digits of an hexadecimal number:

```
const char hexDigits [ ] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F' };
```

```

Void decToHex (unsigned num) {
    const char hexDigits [ ] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F' };
    if (num <= 1) return num;
    int n = num % 16; num /= 16;
    cout << decToHex (num) << hexDigits [n];
}

```

↑
Can't cout a
Void function!

$$\begin{array}{r}
 16 \overline{) 80} \\
 16 \overline{) 5} \\
 0
 \end{array}$$

$16' 16^{\circ}$

$$\begin{array}{r}
 1201 \\
 256 + 32 + 1 = 258 \\
 16 \times 16
 \end{array}$$

$$\begin{array}{r}
 256 \\
 32 \\
 \hline
 258
 \end{array}$$

$$\begin{array}{r}
 16 \overline{) 304} - 89 \\
 16 \\
 \hline
 14
 \end{array}$$

$$\begin{array}{r}
 16 \overline{) 288 - 0} \\
 16 \\
 \hline
 128 \\
 128 \\
 \hline
 0
 \end{array}$$

$$\begin{array}{r}
 16 \overline{) 288 - 0} \\
 16 \\
 \hline
 128 \\
 128 \\
 \hline
 0
 \end{array}$$

$$\begin{array}{r}
 16 \overline{) 288 - 0} \\
 16 \\
 \hline
 128 \\
 128 \\
 \hline
 0
 \end{array}$$

3/3

33
↓

$$\begin{array}{r}
 2 \overline{)33-1} \\
 2 \overline{)16-0} \\
 2 \overline{)8-0} \\
 2 \overline{)4-0} \\
 2 \overline{)2-0} \\
 \end{array}$$

$$\eta = 33\% / 2 = 1$$

17 May 28
N. P. M. H. C. 1911 1860

num 1-1, fl. 121 4

60

NAME: Khaja Anwar Ali Siddiqui

CSCI 689-2

Quiz 8
Closed Book & Notes
(10 points)

Fall 2015

1. Answer each part of this question.

- a) Suppose that you have a ~~vector~~ < int > container called primes. How would you use an iterator to display every element in primes in ascending order? *so*
- b) Assuming that the variable it is an iterator, describe the effect of the expression *it++.
- c) True or false: If c is a nonempty container, calling c.begin() returns an iterator pointing at the first element of that container.
- d) True or false: If c is a nonempty container, calling c.end() returns an iterator pointing at the last element of that container.
- e) True or false: If we use an iterator to display the contents of a container, that iterator must be a constant iterator.

a)

```
set, lind > :: iterator i;  
for (i = primes.begin(); i != primes.end(); i++) cout << *i << endl;
```

*→ *it++ has same effect as that of (*it)++; the value is dereferenced first
then it is incremented.*

c) True.

d) False.

e) False.

2. Write a C++ program (a) to read an English text from the stdin, one word at a time, and store those words in the set < string > container called words, and (b) to display the contents of two-letter words in the words on stdout in alphabetical order.

```
#include <set>
#include <iostream>
using namespace std;
int main() {
    string word; set<string> S;
    cout << "Give words" << endl;
    cin >> word;
    while (cin != EOF) {
        S.insert (word);
        cin >> word;
    }
    cout << "Sorted format" << endl;
    set<string>::iterator i;
    for (i = S.begin(); i != S.end(); i++)
    {
        cout << *i << endl;
    }
}
4.5
* You need to check if (*i).length() == 2
{
    cout << *i;
}
```

NAME: Khaja Anwar Ali Siddiqui

CSCI 689-2

Quiz 9
Closed Book & Notes
(10 points)

19.5
Fall 2015

1. Answer each part of this question.

- a) What declaration would you use to create and initialize each of the following variables:
 - i. A pointer `bp` that points to a Boolean value;
 - ii. A pointer named `pp` that points to a Point with the coordinates (3, 4);
 - iii. A dynamic array called `names` capable of holding 100 C++ strings?
- b) What statements would you use to free the storage allocated in the preceding exercise?
- c) What is a *memory leak* and what is a *dangling reference*?
- d) What is a *destructor*? What is its most important role?
- e) What does it mean for a variable to *go out of scope*?

0) i) `bool *bp;`

ii) `Point *pp = Point(3,4);`

iii) `String *names = new String[100];`

b) `delete bp; ~Point(); delete[] names;`

c) Memory Leak: Any memory that not deleted after the execution of current block is called as "Memory Leak".

Dangling reference:- A pointer that is referencing to a deleted value is called 'Dangling Reference'.

d) A destructor is used to delete the memory allocated for an object. It's usually called once the object is no longer needed. Its important role is to delete the memory allocated & hence avoid memory leak problems.

e) A variable is accessible within the block it is declared. Once the block is executed the memory allocated will be deleted, making the variable inaccessible called out of scope.

2. Write a C++ program with the function `createIndexArray (n)` that allocates a dynamic array of n integers, in which each integer is initialized to its own index, and it returns the address of this array. The `main ()` routine gets the value n from the `stdin`, calls the function `createIndexArray ()`, and prints out the values in the returned array on `stdout`.

```
#include <iostream>

using namespace std;
int* createIndexArray(int); // Function declaration

int main()
{
    int n;
    cout << "Provide the index value: " << endl;
    cin >> n;
    int* address = createIndexArray(n);
    cout << "for (int i=0; i<n; i++) {"
    cout << "    cout << \"Array element at index: \" << i << *address;"
    cout << "    address += 1; }"
}

int* CreateIndexArray(int n)
{
    int* array = new int[n];
    for (int i=0; i<n; i++)
        array[i] = i;
    return array;
}
```

must be a
int → address = createIndexArray (n);
You had it!
Plus you are returning an int* →

4.5/5

NAME: Khaja Anwar Ali Siddiqui

(9)

CSCI 689-2

Quiz 10
Closed Book & Notes
(10 points)

Fall 2015

1. Consider the following statements:

```
class X {  
public:  
    void one();  
    void two(int, int);  
    X();  
private:  
    int a, b;  
};
```



```
class Y : public X {  
public:  
    void one();  
    Y();  
private:  
    int c;  
};
```

Suppose the following statements are in a user program (client code): X x; Y y;

- a) True or false: the private members of X are public members of Y?
b) Mark the following statements as valid or invalid. If a statement is invalid, explain why.
- void X::one() { cout << a + b << endl; }
 - x.a = 15; y.b = 30;
 - void Y::one() { a = 10; b = 15; c = 30; cout << a + b + c << endl; }
 - cout << x.a << ' ' << x.b << ' ' << y.c << endl;

a) False.

b) (i) Valid

(ii) Invalid (The objects x & y will have access to only public members).

x.a = 15; // Invalid (\because a is a private member).

y.b = 30; // Invalid (\because b is a private member of class X),

(iii) Invalid (\because a & b are private to class X & can't be assigned using derived class).

(iv) Invalid (Private members (a,b,c) can't be accessed outside the class).

2. Consider the following statements:

```
class A {
public:
    A ( const unsigned& = 1 ); // constructor, where the argument is
                            // default size for an object
    A ( const A& ); // copy constructor
    A& operator= ( const A& ); // assignment operator
    ~A (); // destructor
private:
    unsigned sz; // size of array
    int* p; // points to integer array
};
```

1. Show the implementation of the constructor. Assign consecutive integers to the elements of the integer array, starting from 1.
2. Show the implementation of the copy constructor.
3. Show the implementation of the assignment operator.
4. Show the implementation of the destructor.

① $A :: A(\text{const unsigned\& } \text{size} = 1)$

```
sz = size;
P = new char [sz];
for (int i=0; i<sz; i++)
    *P[i] = i+1;
```

② $A :: A(\text{const A\& arr})$

```
A temp(arr.size); // new array of sz
int size = arr.size; // storing size
for (int i=0; i<sz; i++) *temp[i] = arr[i];
```

\nearrow Calling your constructor
fills with values.
 \nwarrow You don't need to
call it.

```
P = new char [size];
for (int i=0; i<size; i++) *P[i] = *temp[i]; // copying data
sz = size; // assigning size
```

③ $A \& A::operator=(\text{const A\& arr})$

```
int size = arr.size;
A temp(size); // loop
for (int i=0; i<size; i++) *temp[i] = arr[i];
```

```
return temp;
```

④ $A::NA()$

```
delete [] P;
```

\nearrow In ② & ③ You should set sz & P to
arr.size & loop $P[i] = arr.p[i]$.
(Simpler solution)

NAME: Khoga Anwar Ali Seldique

CSCI 689-2

Quiz 11
Closed Book & Notes
(10 points)

Fall 2015

1. What is the output of the following C++ program?

```
class B {
public:
    B ( string s = " ", int a = 0 );
    void print ( ) const;
protected:
    int x;
private:
    string str;
};
```

```
B :: B ( string s, int a ) : x ( a ), str ( s ) { }
void B :: print ( ) const { cout << x << " " << str << endl; }✓
```

```
D :: D ( string s, int a, int b ) : B ( "Hello Base", a + b ), y ( b ) { }
void D :: print ( ) const { cout << "Derived class: " << y << endl; B :: print ( ); }
```

```
int main ( )
{
    B b ( "This is the base class", 2 ); ✓
    D d ( "DDDDDD", 3, 7 );

    ✓b.print ( ); d.print ( );
    return 0;
}
```

O/P: 2 This is the base class

Derived class: 7

10 Hello Base

ADD, 3, 7

↓
Hello Base, 10, y = 7

x = 10

Pvt:
str = "Hello Base";

d.print()
↓

sc.y

B::print()

10 Hello Base

2. The following class can be used to represent long integers in a computer system:

```
class longInt {  
public:  
    ① longInt ( long = 0, unsigned long = 0 );  
    ② longInt ( const longInt& );  
    ③ longInt& operator= ( const longInt& );  
    ④ ~longInt ();  
    ⑤ void print ( ) const;  
  
private:  
    long msp; // most significant part  
    unsigned long lsp; // least significant part  
};
```

The following is a test program for this class:

```
int main ()  
{  
    longInt x, y ( 1 ), z ( 1, 2 ), u ( z );  
  
    cout << "x = "; x.print ( ); cout << endl;  
    cout << "y = "; y.print ( ); cout << endl;  
    cout << "z = "; z.print ( ); cout << endl;  
  
    cout << "u = "; u.print ( ); cout << endl;  
    u = y; cout << "u = "; u.print ( ); cout << endl;  
  
    longInt v = longInt ( 0xff, 0xffffffff0 );  
    cout << "v = "; v.print ( ); cout << endl;  
    return 0;  
}
```

And the output is:

```
x = ( 0x0, 0x0 )  
y = ( 0x1, 0x0 )  
z = ( 0x1, 0x2 )  
u = ( 0x1, 0x2 )  
u = ( 0x1, 0x0 )  
v = ( 0xff, 0xffffffff0 )
```

Show the implementation of all member functions of the class longInt. Your print() function should print all objects of longInt in hexadecimal form.

quiz - 1 check for self assignment.

```
? if (this != &l) {  
    msp=l-msp; lsp=l-lsp; }  
    return *this;
```

2

Ans

① longInt::longInt(long m, unsigned long l) {
 msp=m; lsp=l; }

② longInt::longInt (const longInt& a) {
 msp=a.msp;
 lsp=a.lsp; }

③ longInt& longInt::operator= (const longInt& m) {
 check for self assignment
 msp=m.msp;
 lsp=m.lsp
 return *this; }

④ longInt::~longInt() {
 msp=0; lsp=0; }

⑤ void longInt::print() const {
 cout << "(" < hex < msp < ", " < hex
 < lsp < ")" < endl;
 ? }

NAME: Khaja

CSCI 589-2

NAME:

Khaja Anwar Ali Siddiqui

9.8

CSCI 689-2

✓ What's
100?

Quiz 12
Closed Book & Notes
(10 points)

Fall 2015

1. Suppose that you have the following classes, A and B:

```
class A {  
public:  
    A ( int = 0 );  
    void f ( );  
    virtual void print ( ) const;  
private:  
    int x; 12, 13  
};
```

```
A :: A ( int a ) : x ( a ) { }  
void A :: f ( ) { x *= 2; } ← 12 - 13  
void A :: print ( ) const {  
    cout << "A :: x = " << x << endl;  
}
```

```
class B : public A {  
public:  
    B ( int = 0, int = 0 );  
    void f ( );  
    void print ( ) const;  
private:  
    int y; 5 (6, 10)  
};
```

```
B :: B ( int a, int b ) : A ( a ), y ( b ) { }  
void B :: f ( ) { A :: f ( ); y *= 2; } ← Not virtual  
void B :: print ( ) const {  
    A :: print ( ); cout << "B :: y = " << y << endl;  
}
```

- (a) What is the output of the following program?

```
int main ()  
{  
    ← A* p; A a ( 2 ); B b ( 3, 5 );  
    p = &a; p->f ( ); p->print ( ); 14  
    p = &b; p->f ( ); p->print ( );  
    return 0; 15  
}  
  
Virtual counts
```

Ans:- A :: x = 4

A :: x = 6 // 3 → 6
B :: y = 5 // 5 → 5

x = 3, y = 5
↓
3 10

- (b) What is the output of the previous program if the definition of class A is replaced by the following definition?

```
class A {  
public:  
    A ( int = 0 );  
    virtual void f ( );  
    virtual void print ( ) const;  
private:  
    int x;  
};
```

Ans

A :: x = 4
A :: x = 6 // 3 → 6
B :: y = 10 // 5 → 10

2. Write a C++ *function template*, named Swap (), that can be used to swap the values of its two arguments. See the following example program for a usage of this function:

```
int main ( )
{
    int x = 5, y = 7; Swap ( x, y );
    string s = "ABC", t = "XYZ"; Swap ( s, t );
    return 0;
}
```

Template <typename T>
Void Swap(T&x,T&y){
 T temp=x;
 x=y;
 y=temp;
}

4.8/5