CS101: Autumn 2017 End Semester Exam

Sat. 18th November 2017, 9.30am-12.30pm

Name: _____**UMESH BELLUR**_____

Roll number: _____----------_____ Hall/Column_____

Lab Group: (Eg: Wed-SL2-2)____**LAB SETTER BATCH**_____**(-2 pts if this is blank/incorrect)**

Write relevant answers ONLY in the space provided - don't ramble. Answers must be written in pen (not pencil) ONLY - else, we will not allow you to crib. Write your roll number on ALL pages. Any minor syntax errors in code snippets are typos - ignore them and assume the right syntax. ***Questions with a * next to the point marking carrying negative marking of 1 mark or as indicated for incorrect answers.***

-5 points for messing with the table below.

Q #	Marks	Grading TA	Q #	Marks	Grading TA	CRIBS (Q # - followed by reason for crib.)
1			16			
2			17			
3			18			
4			19			
5			20			
6			21			
7			22			
8			23			
9			24			
10			25			
11			26			
12			27			
13			28			
14			29			
15			**TOTAL:**			

CS101: Autumn 2017 End Semester Exam

Name: _____ **Roll number:** _____

1. The output of the following program is __0000___ **[1 point - no partial]**

```
int main() {
    static int i=5;
    if(--i){
        main(); // Yes this is legal.
        cout << i;
    }
}
```

2. How many times will the following program print cs101? ____11_____ **[1.5 points - no partial]**

```
int main() {
    int i = 1024;
    for (; i; i >>= 1)
        cout << "cs101";
    return 0;
}
```

3. The output of the following program **is**____0010____ **[2 points - no partial]**

```
int main() {
   int a = 1;
   int b = 1;
   int c = a || --b;
   int d = a-- && --b;
   cout << a << b << c << d;
   return 0;
}
```

4. The output of the following program is __-121___ **[1 point - no partial]**

```
int main() {
    char c = 125;
    c = c+10;
    printf("%d", c);  // %d is to print the contents of c as
                          // an integer.  printf is like cout.
    return 0;
}
```

5. The output of the following program is: ___same___ **[1 point - no partial]** *

```
int  main() {
   unsigned int x = -1;
   int y = ~0; // Note the bit operation here.
   if (x == y)
      cout << "same";
   else
      cout << "not same";
   return 0;
}
```

Name: _____ **Roll number:** _____

6. The following statement is equivalent to which of the options given? **[1.5 points - no partial]** *

```
x  = x | 1 << n;
```

a. Sets x = 2^n
b. Sets the (n+1)th bit of x
c. Toggles the (n+1)th bit of x
d. Unsets the (n+1)th bit of x

7. The value printed by the following program is:__3__ **[2.5 points - no partial]**

```
int f(int *p, int n) {
    if (n <= 1) return 0;
    else return max(f(p+1,n-1),p[0]-p[1]);
}
int main() {
    int a[] = {3,5,2,6,4};
    cout << f(a,5);
}
```

8. If the program compiles, what is its output? If it does not compile, state the line number where the compiler error occurs. **[1 point - no partial]**

```
1.  void foo(int x=0, char y='c'){
2.      cout << "in foo1" << endl;
3.  }
4.
5.  void foo(float x, int y=0){
6.          cout << "in foo2" << endl;
7.  }
8.
9.  int main(){
10.         float x = 3.4;
11.         foo(x);
12.         foo(3.4);
13. }
```

Compiler error at line 12. Call to foo(3.4) is ambiguous

9. Complete the function to swap two integers **without using a temporary variable**. Assume no overflow occurs on arithmetic operations. However, one of the numbers may be zero. **[3 points]**

```
void swap(int& x, int& y){
            // Your code goes here. No more than 5 lines.
            x = x + y;
            y = x - y;
            x = x - y;
}
// ANY OTHER VALID SOLUTION is ACCEPTABLE. As long as it does not do a
divide  by 0.
```

Name: _____ **Roll number:** _____

10. Here is a function that will print out the sum of all odd numbers and the sum of all even numbers from 0 till an upperbound (including the upperbound). Fill in the blanks. **[2 points]**

```
void sumOddEven(unsigned int upperbound){
   int sumOdd  = 0; // For accumulating odd numbers, init to 0
   int sumEven = 0; // For accumulating even numbers, init to 0
  // Use a while-loop to repeatedly add 1, 2, 3,..., to the upperbound
   int number = 1;
   while (number <= upperbound) {

      if (_____number%2 == 0 ____) { // 1.5 points
         sumEven += number;
      } else {
         sumOdd += number;
      }

        _number++;___________ // 0.5 points
  }
  // Print the results
   cout << "The sum of odd numbers is " << sumOdd << endl;
   cout << "The sum of even numbers is " << sumEven << endl;
}
```

11. Rewrite the following code segment as a switch statement. **[2 points]**

```
if (n == 1) {
    total = total + 2;
}
else if (n == 2) {
    total = total + 27;
}
else if (n == 3) {
    total = total + 314;
}
else{
    total = total - 6;
}
```

```
switch (n)                          0.5
{
  case 1:
    total = total + 2;              0.25
    Break;                          0.25
  case 2:                           0.25
    total = total + 27;
    break;
  case 3:                           0.25
    total = total + 314;
    break;
  Default:                          0.25
    total = total - 6;              0.25
}
```

12. Fill in the blank in the definition of the function InRange given below. InRange should return true if its integer argument is between 4 and 27, ***inclusive***. **[1.5 points]**

```
bool InRange(int n) {
    return _____n >= 4 && n <= 27;____ ;  0.5 for n>=4, 0.5 for && and 0.5
for n<-27
}
```

Name: _____ **Roll number:** _____

13. Here is a program that prints the second word of a given sentence and its length. In this case, the sentence is “Actually, couldn’t you come with me?”. The program prints couldn’t(8). Fill in the blanks to make it happen. Assume the existence of a function called isSeparator(char x, char* separators) that returns a boolean indicating whether x belongs to the list of separators. Word separators are tabs, spaces and punctuations. There may be consecutive separators (like in the given sentence there is a comma followed by a space) and in this case the second word is the one following both these separators. **[6 points]**

```
int main() {
	int ret = 0;
	char endOfString = '\0';
	char theSentence[] = "Actually, couldn't you come with me?";
	char separators[] = " \t,.;:?!";// these are all the separators we will recognize.
	char* strPtr = theSentence;
	char* strEnd;
	// find the beginning of the second word
	bool isSecondWord = false;

	while(___*strPtr != '\0' _______) {  // 1.5 points

		while(isSeparator(*strPtr, separators)) {
			// while deals with case with consecutive separators
			strPtr++;
			 isSecondWord = true;
		}
		if (isSecondWord) {
		  // end of consecutive separators, means second word starts at strPtr
			break;
		} else {
			__ strPtr++; _; // 1 point
		}
	 }
	 // find the end of the second word
	strEnd = strPtr;
	while(*strEnd != endOfString) {

		if (___isSeparator(*strEnd,separators)__) {  //2 points

			// first separator found after second word, mark the end
			*strEnd = endOfString;
			break; // 1.5 points
		}
		strEnd++;
	}
	// print the second word and its length
	cout << strPtr << "(" << strlen(strPtr) << ")" << endl;
	return ret;
}
```

Name: _____ **Roll number:** _____

14. Given the following main program, write the declaration (no implementation) of a class **Complex** that will support this main program. You may write non class operators as well and if needed they must be supported by appropriate friend declarations. Your class must have two double class members to represent the real and imaginary parts and both of these must be **private** to the class. ***The class must be minimal in that extraneous members that are not needed for this main program will cost you points***. **[5.5 points]**

```
int main(int argc, char** argv) {
       if (argc < 4) {
            cout << "need an operation on command line" << endl;
            cout << " e.g. ./test 4.3+3i - 2+3.1i" << endl;
            cout << " (no spaces within each complex number, use spaces to\n";
            cout << " separate the operands and the operator - use arithmetic\n";
            cout << " operators only)" << endl; return 1;
       }

      string complexString1 = argv[1];
      string complexString2 = argv[3];
      Complex complex1;
      complex1.fromString(complexString1);
      Complex complex2;
      complex2.fromString(complexString2);
      Complex complex3;
      if (argv[2][0] == '+') {
             complex3 = complex1 + complex2;
      } else if (argv[2][0] == '-') {
            complex3 = complex1 - complex2;
      } else if (argv[2][0] == '*' || argv[2][0] == '.') {
             complex3 = complex1 * complex2;
      } else if (argv[2][0] == '/') {
            complex3 = complex1 / complex2;
      }
      cout << complex1 << " " << argv[2][0] <<
            " (" << complex2 << ") = " << complex3 << endl;
      double d1 = 3.145;
      complex3 = d1 * complex3;
      return 0;
}
```

WRITE YOUR SOLUTION ON THE NEXT PAGE!!

Name: _____ **Roll number:** _____

SOLUTION TO QUESTION 14 HERE

```
class Complex {
    private:                                    1 point for declaring 2 doubles as private members.
        double real;
        double imag;
    public:
        void fromString(const string&);   // 0.5 point if the signature is correct.
        Complex operator+(const Complex&);   // 0.5 point if the signature is correct.
        Complex operator-(const Complex&);  // 0.5 point if the signature is correct.
        Complex operator*(const Complex&);  // 0.5 point if the signature is correct.
        Complex operator/(const Complex&);  // 0.5 point if the signature is correct.

        friend ostream& operator <<(ostream&, Complex&);  // 0.5 point if the signature is correct.
        friend Complex operator *(double, const Complex&);
};

ostream& operator<<(ostream&, Complex&);  // 0.5 point if the signature is correct.
Complex operator * (double, const Complex&);  // 1 point if the signature is correct and friend exists.
```

If the friend is not there, there must be getter/setter methods for real and imag which are public.

Name: _____ **Roll number:** _____

15. Complete the heading for function F so that, on return from the call to F in main, the variable n1 contains the value 3 and the variable n2 contains the value 7. **[2 points - no partial]**

```
void F( _int& k2, int k1_ ) {
    k1++;
    k2 = k2 + 2;
}

int main() {
    int n1 = 3, n2 = 5;
    F(n2, n1);
}
```

16. Complete the given program segment by filling in the blanks that, given a vector of vectors named **vals** and coordinates of one of its elements in **row** and **col**, prints the values that lie on the lower-left to upper-right diagonal of **vals** *that contains the element with the given coordinates*. The values should be printed one per line; the given program prints the values up from row,col to the upper right corner of that diagonal first and then down from row+1,col-1 to lower left corner next. **[5 points]**

```
// print values up from row,col
r = row;
c = col;
while (r >= 0 && _c < vals[0].size()__) // 1.5 points
{
      cout << vals[r][c] << endl;

      r--; // 0.5 point
      c++;
}

// print values down from row,col
r = row + 1;
c = col - 1;

while (c>=0 && r < vals.size() ){ // 0.5 + 1.5 points
    cout << vals[r][c] << endl;

    __r++__; // 0.5 point

     __c--__; // 0.5 point
}
```

Name: _____ **Roll number:** _____

17. Given below is a function to sort a vector of ints.
`void Sort (vector<int> &values){// the actual code does not matter for this question }.`
Rewrite the function as a public member function of a **Collection** class template that represents a collection of objects of the template parameter type. Only the declaration is needed. **[3 points]**

```
template <class T>                          // 1 point
void Collection<T>::Sort ( );               // void sort = 1 point + Collection<T>:: = 1 points
```

OR

```
template <class T>
class Collection<T> {
        public:
                void Sort();
};
```

18. The following code contains three **serious memory bugs** (logical errors, NOT syntax errors), usually ending in a segmentation fault. Find the line number of the bugs, explain what the bug is BRIEFLY and the fixed code. The infinite loop is intentional and meant to illustrate one of the bugs - it's not a bug by itself. **[8 points - 2+4+2]** **// 0.5 point for line number and 0.5 point for fix and rest for the description.**

```
 // this program is buggy
1  int main() {
2     while (true){
3      double* d = new double;
4      for(unsigned int i = 0; i < 3; i++) {
5           d[i] = 1.5 + i;
6      } // end of for
7      for(unsigned int i = 2; i >= 0; i--)
8            cout << d[i] << endl;
9     } // end of while.
10 } // end of program
```

Line No.	Bug Description	Fix
3 or 5	It allocates only one double instead of 3 used in the for loop for assignment. So it overwrites memory.	double* d = new double[3];
7	where we use an unsigned int as a counter to count down. This will be an infinite loop in itself since unsigned ints cannot take negative values needed to get out of the loop. Hence the reading of d[i] starts reading unallocated memory after 3 iterations. This can cause a segmentation fault.	for(int i=0;i<3;i++)
9	Does not delete the doubles allocated in the while(true) loop. MEmory leak	delete d[]; just before line 9

Name: _____ **Roll number:** _____

19. The following code initializes an STL vector of integers to (2, 1) then iterates to erase all elements with value 1. The code compiles but on running we get a segmentation fault abort. Find and fix the logical bug by rewriting the for loop only. **[2+3 points].**

```
// Yet another buggy program. Remember those cute kittens:-))
int main() {
        int ret = 0;
        vector<int> v1;
        v1.push_back(2);
        v1.push_back(1);
        vector<int>::iterator vi;
        for(vi = v1.begin(); vi != v1.end(); vi++) {
                if (*vi == 1) {
                        v1.erase(vi);
                 }
        }
        return ret;
 }
```

The erase of an element in a container being iterated over will change the end of the container.As a result since vi is incremented before the check of vi != v1.end, it will never be equal to it (the end will now be before its value) and hence this results in a infinite loop till a bad memory access happens and a resulting segmentation fault. The fix is:

```
for(vi = v1.begin(); vi != v1.end(); ) {
        if (*vi == 1) {
                vi = v1.erase(vi);
        } else
                vi++;
    }
}
```

The member function erase on template class vector.

```
iterator erase(const_iterator position);
```

Removes from the vector either a single element (*position*).

Parameters

Position - Iterator pointing to a single element to be removed from the vector.

Return value

An iterator pointing to the new location of the element that followed the last element erased by the function call. This is the container end if the operation erased the last element in the sequence.

Name: _____ **Roll number:** _____

20. Consider the following class for a ListNode in a singly linked list.

```
class ListNode {
    public:
      ListNode (int k);   // initialize myValue to k and myNext to 0
      ListNode (int k, ListNode* ptr);//initialize myValue to k and myNext to ptr
      ~ListNode ();       // delete the node and all nodes
                        // accessible through it; precondition: this != 0
      int First ();       // return the value stored in the list node
      ListNode *Rest ();  // return the list consisting of the remaining elements
      void Print ();      // print the list
     private:
      int myValue;
      ListNode* myNext;
};
```

There are **three logical bugs/errors** in the following ListNode member function called **RemoveEvens** that is intended to remove nodes that contain even integers from a nonempty object list and delete the nodes getting removed.

```
void ListNode::RemoveEvens () {
      for (ListNode *temp = this; temp != 0; temp = temp->myNext) {
            if (temp->myNext->value % 2 == 0) {    // even number?
                  delete temp->myNext;
                  temp->myNext = temp->myNext->myNext;
            }
      }
}
```

Identify any two of the three bugs - you don't have to suggest how to fix them. **[3+3 points]**

Bug #	Bug Description
1	**First, the loop termination condition is incorrect; it should be temp->myNext != 0.**
2	**Finally, the node with the even value is deleted too soon; the contents of its myNextmember needs to be saved first, in order to update temp->myNext correctly.**
3	**the value in the first list element is never checked; fixing this requires maintaining a dummy header node in each list, since assignment to "this" is not allowed.**

Name: _____ **Roll number:** _____

21. What is the output of the following program: ___D B2 B1__**[1.5 points - no partial]**

```
class Base1 {
 public:
     ~Base1()  { cout << " B1" << endl; }
};

class Base2 {
 public:
     ~Base2()  { cout << " B2" << endl; }
};

class Derived: public Base1, public Base2 {
   public:
     ~Derived()  { cout << " D" << endl; }
};

int main() {
   Derived d;
   return 0;
}
```

22. What is the output of the following program? _Inside Q___**[0.5 points - no partial]***

```
class P {
 public:
   void print()  { cout <<" Inside P"; }
};

class Q : public P {
 public:
   void print() { cout <<" Inside Q"; }
};

class R: public Q { };

int main(void) {
  R r;
  r.print();
  return 0;
}
```

Name: _____ **Roll number:** _____

23. For the given program, choose one of the options given on the right: **[1.5 points - no partial]** *

```
class Base {
public:
    void show() {cout<<"Base "; }
};

class Derived: public Base {
public:
    int x;
    void show() {cout<<"Derived ";}
    Derived() {x = 10;}
};

int main(void) {
    Base *bp, b;
    Derived d;
    bp = &d;
    bp->show();
    cout << bp->x;
    return 0;
}
```

1. Base 10
2. Derived 10
3. Compiler error in bp->show()
4. **Compiler error in cout << bp->x**

24. Predict the output of the program by picking one of the choices on the right. **[2 points - no partial]** *

```
1. class Test {
2.   public:
3.     static void fun1(){
4.        cout << "Inside fun1()";
5.     }
6.     static void fun2(){
7.        cout << "Inside fun2()";
8.        this->fun1();
9.     }
10.};
11.
12.int main() {
13.  Test obj;
14.  obj.fun2();
15.  return 0;
16.}
```

a. Inside fun2() Inside fun1()

b. Inside fun2()

c. Inside fun1() Inside fun2()

d. Compiler error

If you picked choice 4, state which line the compiler error is at: **8 (cannot access this in a static method)** to complete your answer.
Only a complete answer receives points.

Name: _____ **Roll number:** _____

25. What is the output of the following program? **[4 points]**

```
class A {
        public:
        A() { cout << "A::A " << endl; }
        ~A() {
            cout << "A::~A() " << endl;
            throw “A::exception”;
        }
};
class B {
       public:
       B() {
           cout << "B::B() " << endl;
           throw "B::exception";
       }
       ~B() { cout << "B::~B()"; }
};
int main() {
        try{
              A a1;
              B b1;
              cout << " Exiting “ << endl;
        } catch (const char* ex) {
              cout << ex << endl;
        }
        return 0;
 }
```

A::A // 0.5 point
B::B() // 0.5 point
A::~A() // 1 point
libc++abi.dylib: terminating with uncaught exception of type char const*
// 2 points for detecting that there will be an uncaught exception.

26. What is the output of the following program? **[3 points]**

```
void f1(double t);
 int main() {
       try {
              cout << "stage 1" << endl;
              f1(80);
              cout << "stage 2" << endl;
              f1(40);
               cout << "stage 3" << endl;
       } catch (int x) {
              cout << x << endl;
        }
       cout << "stage 4" << endl;
}

void f1(double t) {
        cout << "Beginning f1" << endl;
       if (t < 50) throw 25;
}
```

stage 1
Beginning f1
stage 2
Beginning f1
25
stage 4

0.5 point per line.

Name: _____ **Roll number:** _____

27. Indicate in each of the blanks in the comments in the code which function will be called (e.g. A::foo1() or B::foo2() etc.,.). If there is an error in any of the lines, write **Error** in the corresponding blank. **[3 points]**

```
class A {
public:
    virtual void foo1() = 0;
    virtual void foo1(int){}
    void foo2(){}
    void foo2(int){}
};

class B: public A {
public:
    void foo1(){}
    void foo2(){}
};

int main() {
    B b;
    int x = 0;

    b.foo1();   //____class B::foo1()_____0.5 points*(-0.5 pts)


    b.foo1(x);  //__error_____1 point*(-1 point)


    b.foo2();   //__class B::foo2()_________0.5 points*(-0.5 pts)


    b.foo2(x);  //___error_______1 point*(-1pt)


    return 0;
}
```

Name: _____ **Roll number:** _____

28. **Given the following code:**

```
class A {
    int a1;
protected:
    double a2;
public:
    void a3();
};
```

```
class B {
    int b1;
protected:
    int b2();
public:
    double b3;
};
```

```
class C: private A, public B { };
```

Fill the table with names of data members and member functions inherited by class C which: **[2 points]**

Cannot be accessed by members of C	**a1, b1**
Have Private accessibility in C (visible only within C)	**a2, a3(),**
Have Protected accessibility in C	**b2()**
Have Public accessibility in C	**b3()**

29. Fill in the missing lines of code to draw the shape shown on the right. Note that the arguments to the function drawPicture can change - so you must express your answers as a function of the parameters where needed. **[4.5 points]**

```
void drawPicture(int sideLength, int numTriangles){

  repeat(numTriangles){ // 0.5 point
    forward(sideLength);
     left(90); // 0.5 point
    forward(sideLength);
    left(135);
     forward(sideLength*sqrt(2));; // 2
points
    left(135);
     left(360/numTriangles); // 1.5 points
   }
}

main_program {
   int lengthOfSide = 200;
   int k = 12;
   turtleSim();
   drawPicture(lengthOfSide, k);
}
```