Introduction to Computer Science I

Zhengjun Liang

May 8, 2018

Contents

1 April 2nd

1.1 Class Information

Class website: `http://cs.ucla.edu/classes/spring18/cs31`
Midterms:
Thu. Apr 26, 5:15-6:20pm or 6:00-7:05pm
Thu. May 24, 5:15-6:20pm or 6:00-7:05pm
Final:
Sat. June 9, 11:30-2:30pm

1.2 Computer Structure

Components: Central Processing Unit (CPU); Memory; Input/Output Devices
ASCII: Some examples: H→ 72; e→ 101; space→ 32; !→ 33; 6→ 54
Unicode: Unifying Chinese, Japanese, and Korean characters
Record Sound: Sample frequencies, close enough to fool human beings' ears
Record Video: 24 frames per second

1.3 More About Memory

Things you can do with memory:

1. Read a value
2. Store a value

If we want to do more, we need some functions beyond that.

2 April 4th

2.1 Machine Language

Accumulator: 00042
Instruction Counter: 000

21: Operation Code. Copy the number at the indicated memory address into the accumulator
11: Add the number at the indicated memory address into the accumulator
22: Copy the number in the accumulator to the indicated memory address
006: Address
99: Halt

Number →  → Result

Different models of computer built have different machine language, so it is difficult to transfer program in one computer to another

Assembly Language: Program is assembled into machine language by an assembler. Every line of assembly language instruction represents a line of machine language instruction.

load price
add fees
store totalcost

halt

2.2 Assembly Language

FORTRAN

```
integer price = 42
integer fees = 13
integer totalcost = price + fees
```

2.3 Higher-Language

Program is *compiled* into machine language by a compiler.

2.3.1 Programming Languages

C, C++, Objective-C, Java, C#, Python, Perl, Ruby

C++

- C++ is created by Bjarne Stroustrup. Started 1980, made available to the world in 1985
- C++ has "dialects", different compilers may give slightly different results
- 1998 ISO C++ Standard (C++ 98)
- 2011 revision (C++ 11)
- 2014 revision (C++ 14)
- 2017 revision (C++ 17)
- 2020 revision (C++ 20)

3 April 9th

3.1 Errors

Compilation Error: "Beat 3 eggs into a mixing bowl" translated to "into eggs mixing 3 bowl a"
Logic/Runtime Error: "Beat 3 eggs into a mixing bowl" translated to "beat 3333 eggs into a mixing bowl"
The difference between $f(x, y) = x^2 + y^2$ and $f(x, y) = x * x + y * y$;
distanceFromOrigin (x, y) = sqrt $(f(x, y))$

3.2 A Simple C++ Project

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

int main()
{
    cout << "Hello" << end1;
}
```

The output will be:

```
Hello
```

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

int main ()
{
    cout << "How many hours do you work?" >>;
    double hoursWorked;
    cin >> hoursWorked;

    cout << "what is your hourly rate of pay";
    double payRate;
    cin >> payRate;

    cout.setf(ios::fixed); //want fixed point notation
    cout.precision(2); //want the number of digits after the decimal point to be 2

    cout << "You earned $" << (hoursWorked * payRate) << endl;
    cout << "$" << (0.10 * hoursWorked * payRate) << "will be withheld." << end1;
```

The output will be like this:

```
How many hours do you work?
What is your hourly rate of pay? 16.13
You will earn $645.20
$64.52 will be withheld
```

Types of identifiers:

int: -2 million to 2 million

double: $\pm 10^{-308}$ to 10^{308}, about 15 or 16 significant digits

4 April 11th

4.1 Arithmetic Expression

$\boxed{* \ / \ + \ -}$, which might be different from convention in algebra.

Example 1: (3+4)(6-2) versus (3+4)*(6-2). Similarly, division is not written as $\frac{1}{2}$ but 1/2
Example 2: 27/3*3=27, according to the rule of precedence.
Example 3: 14.3/5.0 $\implies$ 2.86; 14/5.0 $\implies$ 2.8; 14/5 $\implies$ 2; 14 % 5 $\implies$ 4 (modular)

```
int a = 0;
int b = a*a;
int c = 25/(b-100); //Undefined Behavior'

double d; //uninitialized variable
double e = a*a;
cout << e;

int f = 1000;
int g = f * f * f;
int h = f * g; //problem
```

4.2 A Weird Behavior

The program:

```
What is your name?
What is your quest? To seek the holy grail
Hello, brave Sir Robin?
You want to seek the Holy Grail.
If you live, next year you will be 33;
```

```
==========================================================================================================
```

```
#include <iostream>
#include <string> //pull in string library
using namespace std;

int main()
{
    cout << "what is your name?";
    string personsName;
    getline(cin, personsName); //we are not gonna do cin >> personsName
    //since it will ignore the blank space
    //we use getline to read strings

    cout << "How old are you?";
    int age;
    cin >> age; //getline only reads strings
    cin.ignore(10000, '\n'); //throw away the rest of the line
    //cin doesn't read \n
    cout << "What is your quest?";
    string quest;
```

```
    getline(cin, quest);

    cout << "Hello, brave" << personsName << "!" << endl;
    cout << "You want" << quest << endl;
    cout << "If you live, next year you will be " << age+1 << endl;
}
```

4.3 A Chart that explains the behavior

	What is written to screen	What the Operation System Holds	Available to the program
You type 3	3	3	
You type 5	5	35	
You type Backspace	BS, Space, BS	3	
You type 2	2	32	
You type enter	CR, LF		32 newline

5 April 13th

5.1 Trivia

C: \BLAH\FOO.TXT

5.2 Statements

5.2.1 If Statement

```
if (payRate >= 14.00) //Be careful of the equal sign; boundary error
   cout << "$" << (0.10 * amtEarned) << "will be withheld." << end1;
else
   cout << "$" << (0.05 * amtEarned) << "will be withheld." << end1
```

is greater than	$>$
is less than	$<$
at least	$>=$
at most	$<=$
not equal to	$!=$
equal to	$==$

5.2.2 Compound statement

{stmt; stmt; stmt;}

```
cout << "What is your name?";
string name;
getline(cin, name);
if (name == "")
   cout << "You didn't enter a name!" << end1;
else
   cout << "Hello, " << name << end1;
```

5.2.3 Declaration Statement

```
type identifier;
type identifier = expression;
```

5.2.4 Assignment Statement

```
variable = expression;

int main ()
{
   double PAYRATE_THRESHOLD = 14.00;
   double HIGH_WITHHOLDING_RATE = 0.10;
   double LOW_WITHHOLDING_RATE = 0.05;

   PAYRATE_THRESHOLD = 15.00; //Error! Won't compile! PAYRATE_THRESHOLD is const!

   double amtEarned = hoursWorked * payRate;
   cout.setf(ios::fixed);
   cout.precision(2);
```

```
   cout << "You earned $" << amtEarned << end1;

   double withholdingRate;
   if (payRate >= 14.00)
      withholdingRate = 0.10;
   else
      withholdingRate = 0.05;

   cout << "$" << (withholdingRate * amtEarned) << " will be withheld." << end1;

}
```

6 April 16th

6.1 More on If Statement

```
string citizenship;
int age;

...

if (citizenship == "USA"){
   if (age >= 18){
      cout << "You can vote in US elections" << endl;
   }
}
else{
   cout << "You are not a U.S. Citizen" << endl;
}

if (expression1 && expression2) //and
if (expression1 || expression2) //or

if (b!=0 && d!=0 && a/b+c/d>10) //fine
if (a/b+c/d>10 && b!=0 && d!=0) //won't compile, cuz order matters
if (18 <= age <= 20) //always true

switch (choice)
{
   case 1:
    ...
    break;
   case 2:
   case 4:
    ...
    break;
   case 3:
   case 5:
    ...
    break;
```

```
   default:
    ...

}
```

7 April 18th

7.1 More on Switch Statement

```
cout << "How many people? ";
int numPeople;
cin >> numPeople;

switch (numPeople)
{
   case 0:
      cout << "Nobody" << endl;
      break;
   case 1:
      cout << "One lonely person" << endl;
      break;
   case 2:
      cout << "A happy couple" << endl;
   case 3:
   case 4:
      cout << "A few people" << endl;
   default:
      cout << "A lot of people" << endl;
}

switch (numPeople)
{
   ...
   case 10: case 11: case 12:
   ...
}

How many times do you want to be greeted? 3
Hello
Hello
Hello

cout << "How many times do you want to be greeted? ";
int nTimes;
cin >> nTimes;
if (nTimes >= 1)
   cout << "Hello" << endl;
if (nTimes >= 2)
   cout << "Hell"
```

7.2 While Loop

```
int n = 0;
while (n < nTimes)
{
   cout << "Hello" << endl;
   n = n + 1;
}
cout << "Good to see you." << endl;
```

Be aware of "off-by-one error";
starting situation;
stay-in-loop condition;

7.3 For Loop

```
for (initialization; stay-in-loop-condition; prepare-for-next-iteration)
   statement

for (int n = 10; n >= 0; n++)
{
   cout << n << endl;
}

for (int k = 1; k< 1000; k *= 2)
   cout << k << endl;
```

7.4 Do-While Loop

```
do
   statement
while (condition);

****
****
****

for (int r = 1; r <= 3; r++)
{
   for (int c = 1; c <= 4; c++)
   {
      cout << "*";
   }
   cout << endl;
}

string s = "Hello" ;
for (int k = 0; k != s.size(); k++){
   cout << s[k] << endl;
}
```

8 April 23rd

8.1 More about String and Char

```
#include <cctype> //whether a char is a digit
int num = 0;
for (int k = 0; k != t; k++){
    if (t[k] == 'o' || t[k] == 'O'){
    num++;
    }
}
cout << The number of O's (upper and lower case) is "<< num << endl;

char c = 'x'; //OK
char c = "x" // Error won't compile
string c = "x" // OK
string c = 'x' //error won't compile


cout << "Enter a phone number: ";
string num;
getline(cin, num);

int numOfDigits = 0;
...
if (numOfDigits != 10){
    cout<< "Error!" << endl;
}


for (int k = 0; k != num.size(); k++){
    if (isdigit(num[k]){
       num++;
    }
}
if (num != 10){
   cout << "A phone number must have 10 digits." << endl;
}

string s;
getline (cin, s);
if (s != ""){
    s[0]=toupper(s[0]);
}

cout << toupper(s[0]);
s[0] = toupper(s[0]);
char c = toupper(s[0]);

if (toupper(t[k] == "0"){
   num++;
}
```

```
string s = "mary";
toupper(s); //error! s is a string
toupper(s[0]) //doesn't do anything useful
```

8.2 Functions

```
void greet(){
   for (int k = 0; k < 3; k++){
      cout << "Hello" << endl;
   }
}

int main (){
   greet ();
}

void greet (int n){
   for (int k = 0; k < n; k++){
      cout << "Hello" << endl;
   }
}
```

9 Apr 25th

9.1 More about Functions

```
int main(){
    ...
    ...
    greet(3, hello);
    ...
    ...
    int a = 4;
    int b;
    cin >> b; //suppose the user typed 2
    greet(a+3*b,"Ni Hao");
    ...
    ...
    ...
    string ml
    getline(cin, m); // suppose the user typed salaam
    if (n != "")
        m[0] = toupper(m[0]);
    greet(2, m);
    ...
    ...
}

int cube(int n){
```

```
    return n = square(n);
}

int square(int n){
    return n*n;
}

void greet (int n, string msg)
{
   if (n < 0){
       cout << "I can't greet you a negative number of times" << endl;

   }
   for (int k = 0; k < n; k++)
       cout << msg << endl;
}
```

10 April 30th

10.1 Functions Continued

```
bool isValidPhoneNumber(string pn);
string digitsOf (string pn);

int main(){
    cout << "Enter a phone number: ";
    string phone;
    getline(cin, phone);
    if(isValidPhoneNumber(phone))
       cout << "The digits in the number are " << digitsOf(phone) << endl;
    else
       cout << "A phone number must have 20 digits." << endl;
}

bool isValidPhoneNumber(string pn){
   int numberOfDigits = 0;
   for (int k = 0; k != pn; k++){
      if (isdigit(pn[k])){
        numberOfDigit++;
      }
   }
   if (numberOfDigits == 10)
      return true;
   else
      return false;
}

string digitsOf (string pn){
   string digitsOnly = "";
   for (int k=0; k != pn.size(); k++){
      if (isdigits(pn[k])
```

```
         ...append pn[k] to the end of digitsOnly ...
   }
   return digitsOnly;
}

void polarToCartesian(double rho, double theta, double xx, double yy);

int main (){
   double r;
   double angle;
   ...//get values from r and angle
   double x;
   double y;
   polarToCartesian(r, angle, x, y);
   ...
   double x2;
   polarToCartesian(2*r, angle, x2, y);
   ...
}

void polarToCartesian(double rho, double theta, double xx, double yy);
{
   xx = rho * cos(theta);
   yy = rho * sin(theta);
}
```

10.2 Pass by Value and Pass by Reference

Reference: A reference is another name for an already existing object.

11 May 2nd

11.1 Arrays

```
void doDateStuff()
{
    const int NMONTHS = 12;

    const int daysInMonth[NMONTHS] = {
        31, 28, 31, 30, 31, 30,
        31, 31, 30, 31, 30, 31
    };

    const string monthName[NMONTHS] = {
        "January", "February", "March",  ...
    };

    ...
    cout << "These months have 31 days:" << endl;
    for (int k = 0; k < NMONTHS; k++)
    {
```

```
        if (daysInMonth[k] == 31)
            cout << monthName[k] << endl;
    }
}
```

```
=======================================================
```

```
int main()
{
    const int MAX_NUMBER_OF_SCORES = 10000;
    int scores[MAX_NUMBER_OF_SCORES];
    int nScores = 0;
    int total = 0;
    cout << "Enter the scores (negative to quit):" << endl;
    for (;;)
    {
        int s;
        cin >> s;
        if (s < 0)
            break;
        if (nScores == MAX_NUMBER_OF_SCORES)
        {
            cout << "I can handle only " << MAX_NUMBER_OF_SCORES
                 << " scores!" << endl;
            cout << "Continuing with just the first "
                  << MAX_NUMBER_OF_SCORES << " values." << endl;
            break;
        }
        total += s;
        scores[nScores] = s;
        nScores++;
    }
    if (nScores == 0)
        cout << "There were no scores, so no stats" << endl;
    else
    {
        double mean = static_cast<double>(total) / nScores;
        cout << "The mean is " << mean << endl;
        double sumOfSquares = 0;
        for (int k = 0; k < nScores; k++)
        {
            double diff = scores[k] - mean;
            sumOfSquares += diff * diff;
        }
        cout << "The std. deviation is "
              << sqrt(sumOfSquares / nScores) << endl;
    }
}
```

```
=============================================================
```

```
double computeMean(const int a[], int n);
void setAll(int a[], int n, int value);

int main()
{
    ...
    const int MAX_NUMBER_OF_SCORES = 1000;
    int scores[MAX_NUMBER_OF_SCORES];
    int nScores = 0;
    ... // fill up the array (perhaps partially)
    double m = computeMean(scores, nScores);

    int stuff[100];
    ... // fill up the entire stuff array
    double m2 = computeMean(stuff, 100);
    ...
    setAll(stuff, 10, 42);
}

double computeMean(const int a[], int n)
{
    if (n <= 0)
        return 0;
    int total = 0;
    for (int k = 0; k < n; k++)
        total += a[k];
    return static_cast<double>(total) / n;
}

void setAll(int a[], int n, int value)
{
    for (int k = 0; k < n; k++)
        a[k] = value;
}
```

12 May 7th

12.1 Two-dim Arrays

```
const int WEEKS = 5;
const int NDAYS = 7;
int attendance[weeks][days];
attendance[2][5]=140;
...
...
for (int w = 0; w < WEEKS; w++){
   int t = 0;
   for (int d = 0; d < NDAYS; d++){
   t = attendance[w][d];
   }
   cout <<"The total for week "<< w << " is " << t << endl;
```

```
}

const string dayNames[NDAYS]={
   "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"
};
int grandTotal;
for (int d = 4; d < NDAYS; d++){
   grandTotal = 0;
   int t = 0;
   for (int w = 0; w < WEEKS, w++){
      t = attendance;
   }
   cout <<"The total for "<< dayTime[d] << " is " << t << endl;
   grandTotal = t;
}
cout <<"Over the course of "<< NWEEKS << " is " << t << endl;
```