1. (9%) Consider the following programming languages: C, C++, Java, Ruby, Python.
   a. (3%) Which languages do not require a compiling stage before execution?
   b. (3%) Which languages support object-oriented programming?
   c. (3%) Which languages are supported with compiler (or interpreter) on *both* Windows and Linux operating systems?

2. (6%) Write down the output of the following C program

```
void reset_array(int a[])
{
  for (int i = 0; i < sizeof(a)/sizeof(a[0]); i++) a[i]=0;
}

void main()
{
  int a[]={1,2};
  reset_array(a);
  for (int i = 0; i < sizeof(a)/sizeof(a[0]); i++)
    printf("%d ",a[i]);
}
```

3. (10%) Write down the output of the following C program

```
void main()
{
  int s[5]={2, 10, 5, 3,7};
  int *p=s , *ptr=s+2 ;
  printf("A:%d\n", *p+1);
  printf("B:%d\n", *ptr);
  printf("C:%d\n", s[0]);
  printf("D:%d\n", *p++);
  printf("E:%d\n", *p);
  printf("F:%d\n", (*p)--);
  printf("G:%d\n", ++*p);
  printf("H:%d\n", *++p);
  printf("I:%d\n", *(s+2));
  printf("J:%d\n", --s[0]);
}
```

4. (4%) List **two** differences between a function prototype and a function header.

5. (8%) Is the following class declaration correct? If not, clearly indicate what lines are wrong and **why** they are wrong.

```
class M {                                                   // 1
  private:                                                  // 2
    int value;                                              // 3
                                                            // 4
  public:                                                   // 5
    M (int i)                { value = i; }                 // 6
    void showValue () { cout << value << endl; }            // 7
                                                            // 8
    void fn1 (int& input)               { value = input; }  // 9
    void fn2 (const int& input)         { value = input; }  // 10
    void fn3 (int& input) const         { value = input; }  // 11
    void fn4 (const int& input) const { value = input; }    // 12
                                                            // 13
    void fn5 (int& input)               { input = value; }  // 14
    void fn6 (const int& input)         { input = value; }  // 15
    void fn7 (int& input) const         { input = value; }  // 16
    void fn8 (const int& input) const { input = value; }    // 17
};                                                          // 18
```

6. (7%) Is the following program correct? If not, clearly indicate all the lines in the code that are not correct and also **what is wrong** with them. If the compiling errors are fixed, what is the output of **line 29**?

```
class Parent {                                              // 1
  private:                                                  // 2
    int a, b;                                               // 3
  protected:                                                // 4
    int c, d;                                               // 5
  public:                                                   // 6
    int e, f;                                               // 7
                                                            // 8
    Parent () {a = b = c = d = e = f = 0; }                 // 9
    void showValues() {                                     // 10
        cout << a << “ “ << b << “ “ << c << “ “            // 11
                << d << “ “ << e << “ “ << f << endl; }     // 12
    friend void foo(Parent);                                // 13
};                                                          // 14
                                                            // 15
class Child : public Parent {                               // 16
    Child ()     { a = b = c = d = e = f = 0; }             // 17
    void fn1 () { a = 2; b = 3; }                          // 18
    void fn2 () { c = 4; d = 5; }                          // 19
    void fn3 () { e = 6; f = 7; }                          // 20
};                                                         // 21
                                                           // 22
void foo (Parent p) {                                      // 23
    p.a = p.b = p.c = p.d = p.e = p.f = 10; }              // 24
                                                           // 25
int main() {                                               // 26
    Parent *p = new Child();                               // 27
    foo(*p);                                               // 28
    p->showValues();                                       // 29
}                                                          // 30
```

7. (6%) What is the output of the following program?

```
class A {
  public:
    A ( ) { cout << "In A's constructor" << endl; }
    ~A ( ) { cout << "In A's destructor" << endl; }
};

class B : public A {
  public:
    B ( ) { cout << "In B's constructor" << endl; }
    ~B ( ) { cout << "In B's destructor" << endl; }
};

class C : public B {
  public:
    C ( ) { cout << "In C's constructor" << endl; }
    ~C ( ) { cout << "In C's destructor" << endl; }
};

int main() {
  C x1;
  C *x2 = new C;
}
```

8. (5%) Suppose we are to insert the following data in the given order, D, M, Q, A, E, K, to build an AVL tree. Please show the final AVL tree,

9. (10%) Please write a C function to insert a new node in the tail of a linked list.
Assume that a data node has the following data structure:

```
struct lnode {
        char key[MaxKeyLen];
        struct lnode *next;
}
```

The input to the function is a string.

10. (10%) Please write a C program to evaluate a postfix expression.
The expression contains two kinds of operators: + and *
The operands are positive integers.
The expression will be given in the command line arguments.
For example,
posteval "10 20 5 * + "

11. (4%) What is the lower bound of the comparison based sorting algorithm? Justify your answer.

12. (5%) The following table shows the used alphabets and their frequencies of appearance (happen to be prime number series) when encrypting a piece of information. Please use greedy algorithm to find its optimal ternary Huffman code for transmission.

alphabet	a	b	c	d	e	f	g	h
frequency	3	5	7	11	13	17	19	23

13. (6%) Analyze each of (1) Quicksort, (2) heap sort, and (3) Merge sort to determine if they are asymptotically optimal or not.

14. (4%) Given an array A with *n* elements that only the adjacent elements might be out of order—i.e., if $i < j$ and A[*i*] > A[*j*], then we will know that $j = i + 1$. To sort A, which of Insertion-sort or Merge-sort is better or equivalent in terms of the time cost? Justify your answer.

15. (6%) Except the running time, what are the main differences between Dijkstra, Bellman-Ford, and Floyd-Warshall algorithms?