

# Program Design

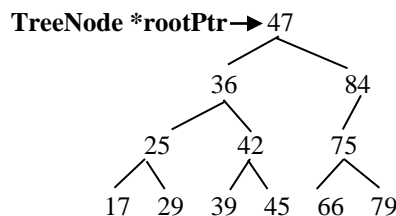
## Final Test

OPEN BOOK and ONLINE

6/22/2021

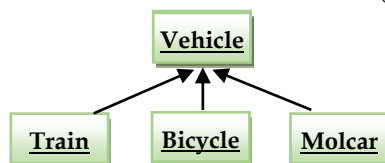
- I. In traditional C, a node of a data structure is usually defined as a self-referential structure. (Total 20 %)
- Please **define** in C a structure **ListNode** for a **doubly linked list** with integer node.
  - Please **define** a **ListNode \*getNewNode(int value)** function and **implement** it to return a NULL pointer if heap runs out of space. Set both pointers to NULL.
  - Suppose a **doubly linked list** is pointed to by the **ListNode \*FirstPtr**, implement the function **void InsertatFront(int val)** that adds a new node to the head of the **doubly linked list**. Use the **getNewNode** function you defined in b.
  - Suppose the last node of a **doubly linked list** is pointed to by the **ListNode \*LastPtr**, implement the function **void InsertatBack(int val)** that adds a node to the end of the **doubly linked list**. Use the **getNewNode** function you defined in b.

- II. A **binary search trees** containing **integers** is built as in the figures below. (Total 25%)



- Please **define** in C++ both classes **TreeNode** and **Tree** for integer binary trees as above. Do not use template class. (10%)
- Please manually travers the binary search tree above using the three traversal algorithms and write out the sequence of numbers visited. (5%)
- How do you generate the sorted sequence in decending order of the numbers contained in the above binary tree. (5%)
- Please manually **insert** the following new node **16, 27, 92, 38, 58, 46, 69, 80, 83** into the binary search tree. Draw the entire final binary search tree. (5%)

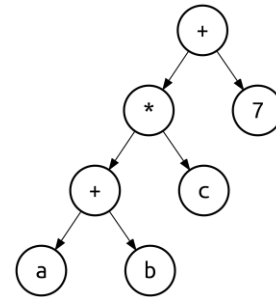
- III. A vehicle class hierarchy is as follows, where Train, Bicycle, and Pui Pui Molcar inherit the abstract class vehicle: (Total 25%)



- Define** and **implement** the vehicle inheritance classes hierarchy. The abstract class **vehicle** and all the concrete subclasses have a method **sound()** to print a sound string, "**making a sound**" for vehicle, "**Bun Bun**" for Trains, "**Lin Lin**" for Bicycle, and "**Gi Gi**" for Molcar. Implement all the classes so that they can demonstrate polymorphism among the concrete subclasses when calling the **sound()** method. (10%)

- Write a program segment in which creating a **vector** of **vehicle** pointers, creating objects of a Train, a Bicycle, and a Molcar and putting them into the vector, and creating a loop with **iterator** to print out the sound strings of all the animal objects in the vector by calling their **sound()** method. (Remember all the criteria for polymorphism) (15%)

- IV. A **binary expression tree** is a specific kind of a binary tree used to represent expressions with both unary and binary operators, e.g., the expression: **(a+b)\*c+7** is represented as: (Total 30 %)



The construction of an **expression tree** is by reading the **postfix expression** one symbol at a time. If the symbol is an operand, a one-node tree is created, and its pointer is pushed onto a stack. If the symbol is an operator, the pointers to two trees T1 and T2 are popped from the stack and a new tree whose root is the operator and whose left and right children point to T2 and T1 respectively is formed. A pointer to this new tree is then pushed to the Stack.

- Modify the **infix** traversal function of the binary search tree from the course so that you can generate **(a+b)\*c+7** back from the expression tree. (10 %)
- For the following one-digit expression:  
 $(9+5)*6-(3+5)/8+7*(8+2)/4-(6+4)/5$   
convert it to the **postfix expression** manually first and use it to generate the corresponding **expression tree**. Write the result postfix expression and draw the final **expression tree**. (20 %)

**NOTE:** Because of the open book evaluation, all the answers to the questions must be precise enough to get the corresponding points.

