PROGRAMMING WITH GENERIC TYPES

15-123

Systems Skills in C and Unix

Generics

- Generic algorithms
  - Algorithms that works on any data type
  - qsort
    - Provide a function pointer that can compare

- Generic Data Structures
  - Generic LL for example
    - Provide function pointers to
      - Compare elements
      - Print elements
      - Free elements

Generics in Java

```
public class Item<T> {
    private T t; // T for type is late binding
    public void init(T t) {
            this.t = t;
    }
    public T get() {
        return t;
    }
}
```

- Item<Integer> intItem;
- Item<Double> doubleItem;

Generics in C++

- **In C++, a Standard Template Library** (STL)
- #include <vector>
  #include <string>

  using namespace std;

  main()
  {
    vector<string> V;

    V.push_back("10");
    V.push_back("20");
    V.push_back("30");   V.pop_back();

    cout << "**Loop by index**:" << endl;

    int i;
    for(i=0; i < V.size(); i++)
    {
      cout << V[i] << endl;
    }
  }

Generics in C

Example: Define a generic linked list class. Start with a generic node.

```
typedef struct LIST_ELEM
{
  void * data;
  struct  LIST_ELEM  * next;
} LIST_ELEM
```

LL Example

define the LL structure

```
typedef struct LINKED_LIST
{
  LIST_ELEM  * head ;
  int (*cmpData)( void *, void *);
  void (*printData)( void *);
  void (*freeData)( void *);
} LINKED_LIST;
```

Design Assumptions

- Cmp, free and print are all acting on the data inside each LIST_ELEM

- LINKED_LIST packages things together to make sure same print, free, print functions apply to all LIST_ELEM

What do function pointers do?

- **Sample Function pointers used in LL struct**
  - **int (*cmpData)(void *, void *);**
  - **void (*printData)(void *);**
  - **void (*freeData)(void *);**
- Use of Function pointers in struct
  - Allow users to define how to compare, print or free data based on data type and pass them to LL structure
  - These functions can be defined during runtime (as part of the specific main program)
    - Stringmain.c
    - Intmain.c

What is the role of each function pointer?

- **What is generic is the data field inside the LIST_ELEM**
- **int (*cmpData)(void *, void *);**
  - **Given two data fields, it defines how to compare that data. For example, comparing ints are different from comparing strings**
- **void (*printData)(void *);**
  - **Given a node, this defines how to print data inside that node.**
- **void (*freeData)(void *);**
  - **Given a node, this defines how to free data inside that node**

Generics Functions defined on the LL structure

```
void initList( LINKED_LIST * list, int (*cmpData)( void *,
  void *), void (*printData)( void *), void (*freeData)( void *) )
```

Generics Functions defined on the LL structure

```
void freeAll(LINKED_LIST * list);
/* free ALL dynamic memory in this program */
```

Generics Functions defined on the LL structure

```
void insertToFront(LINKED_LIST * list,  void *data);
```

Generics Functions defined on the LL structure

void removeAtFront(LINKED_LIST * list)

Questions

based on design of the data structure

- Who allocates memory for LINKED_LIST?
- Who allocates memory for each LIST_ELEM?
- Who frees memory of each LIST_ELEM?
- Who frees memory for LINKED_LIST?

Coding Examples