```
Using a Generic Linked List

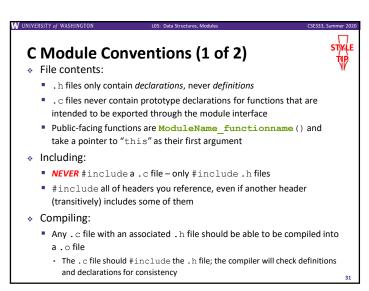
  Type casting needed to deal with void* (raw address)

 Before pushing, need to convert to void*

  Convert back to data type when accessing

 ypedef struct node st {
  void* element:
  struct node_st* next;
1 Node:
 Node* Push (Node* head, void* e); // assume last slide's code
 int main(int argc, char** argv) {
 char* hello = "Hi there!";
  char* goodbye = "Bye bye.";
  Node* list = NULL;
  list = Push(list, (void*) hello);
  list = Push(list, (void*) goodbye);
  printf("payload: '%s'\n", (char*) ((list->next)->element) );
                                              manual list void.c
```

25



W UNIVERSITY of WASHINGTON
 C Header Files
 ★ Header: a file whose only purpose is to be #include'd
 ■ Generally has a filename . h extension
 ■ Holds the variables, types, and function prototype declarations that make up the interface to a module
 ★ Main Idea:
 ■ Every name . c is intended to be a module that has a name . h
 ■ name . h declares the interface to that module
 ■ Other modules can use name by #include-ing name . h
 ▶ They should assume as little as possible about the implementation in name . c

30

