

Functions: Your own library

Computer Programming 143 – Lecture 10

Functions II

Faculty of Engineering



5.1-5.6 Review of Functions III

Function definition format

```
return_value_type function_name( argument_list ) {  
    declarations;  
    statements;  
    return control;  
}
```

- Example:

```
int maximum( int x, int y, int z ) {  
    int max = x;           // assume x is largest  
    if ( y > max )         // if y > max, then max = y  
        max = y;  
    if ( z > max )         // if z > max, then max = z  
        max = z;  
    return max;           // max is largest value  
}
```

Defining your own library

```
mylibrary.h
```

```
// Function prototypes
```

```
mylibrary.c
```

```
#include "mylibrary.h"
```

```
// Function definitions
```

```
main.c
```

```
#include <stdio.h>
```

```
#include "mylibrary.h"
```

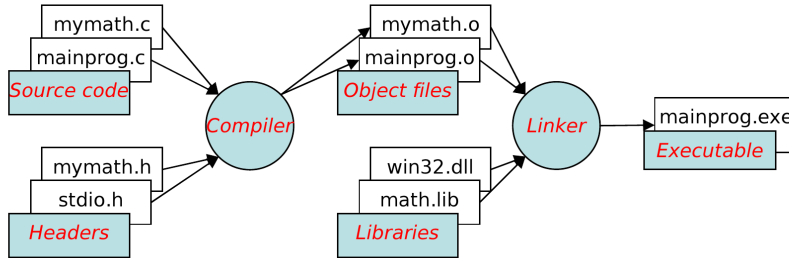
```
int main ( )
```

```
{
```

```
    // Call function
```

```
}
```

5.8 Header Files II



5.8 Header Files III

Standard library header	Explanation
<code><assert.h></code>	Contains macros and information for adding diagnostics that aid program debugging.
<code><ctype.h></code>	Contains function prototypes for functions that test characters for certain properties, and function prototypes for functions that can be used to convert lowercase letters to uppercase letters and vice versa.
<code><errno.h></code>	Defines macros that are useful for reporting error conditions.
<code><float.h></code>	Contains the floating point size limits of the system.
<code><limits.h></code>	Contains the integral size limits of the system.
<code><locale.h></code>	Contains function prototypes and other information that enables a program to be modified for the current locale on which it is running. The notion of locale enables the computer system to handle different conventions for expressing data like dates, times, dollar amounts and large numbers throughout the world.

5.8 Header Files IV

Standard library header	Explanation
<code><math.h></code>	Contains function prototypes for math library functions.
<code><setjmp.h></code>	Contains function prototypes for functions that allow bypassing of the usual function call and return sequence.
<code><signal.h></code>	Contains function prototypes and macros to handle various conditions that may arise during program execution.
<code><stdarg.h></code>	Defines macros for dealing with a list of arguments to a function whose number and types are unknown.
<code><stddef.h></code>	Contains common definitions of types used by C for performing certain calculations.
<code><stdio.h></code>	Contains function prototypes for the standard input/output library functions, and information used by them.
<code><stdlib.h></code>	Contains function prototypes for conversions of numbers to text and text to numbers, memory allocation, random numbers, and other utility functions.
<code><string.h></code>	Contains function prototypes for string processing functions.
<code><time.h></code>	Contains function prototypes and types for manipulating the time and date.



Random numbers

rand()

- In stdlib.h
- Generates a number between 0 and RAND_MAX (~~32767~~)
- How do you generate a number in a specific range?

$$7 \div 5 = 1 + \frac{2}{5} = \frac{7}{5} = \frac{5}{5} + \frac{2}{5}$$

= 1 remainder 2

$$9 \div 5 = \frac{9}{5} = \frac{5}{5} + \frac{4}{5}$$

$$9 \% 5 = 4$$

some big number

0, 1, 2, 3, 4

rand() % 4;

$$7 \% 5 = 2$$

$$105 \% 5 = 0$$

$$108 \% 5 = 3$$

$$1109 \% 5 = 4$$

$$1110 \% 5 = 0$$

$$1114 \% 5 = 4$$

$$32764 \% 5 = 4$$

All between 0 and 4

1, 2, 3, 4, 5

CS 188: Artificial Intelligence

Markov Decision Processes



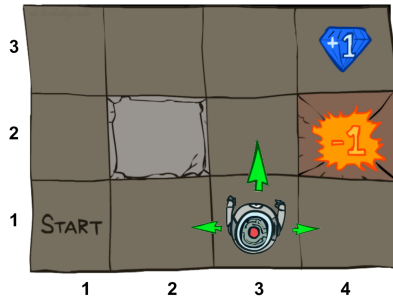
Instructor: Pieter Abbeel

University of California, Berkeley

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley (<http://ai.berkeley.edu>).]

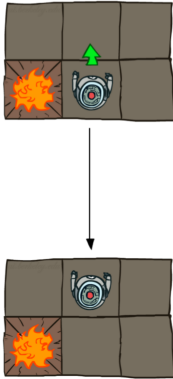
Example: Grid World

- A maze-like problem
 - The agent lives in a grid
 - Walls block the agent's path
- Noisy movement: actions do not always go as planned
 - 80% of the time, the action North takes the agent North (if there is no wall there)
 - 10% of the time, North takes the agent West; 10% East
 - If there is a wall in the direction the agent would have been taken, the agent stays put
- The agent receives rewards each time step
 - Small "living" reward each step (can be negative)
 - Big rewards come at the end (good or bad)
- Goal: maximize sum of rewards

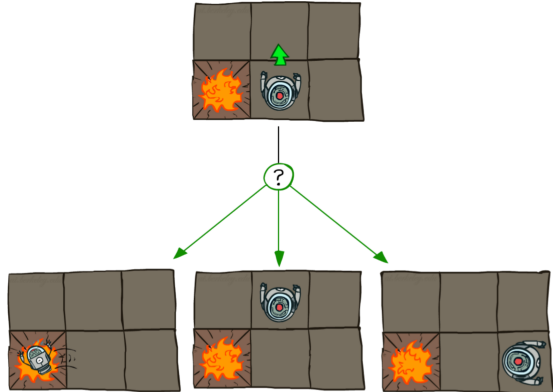


Grid World Actions

Deterministic Grid World



Stochastic Grid World



INTRODUCTION TO **COMPUTER PROGRAMMING**

Brian Reffin Smith

Edited by Lisa Watts

Designed by Kim Blundell

Illustrated by Graham Round
and Martin Newton

