Asymptote Reference Card

Program structure/functions

import "filename" import module
import "filename" as name import variable declaration
include "filename" include verbatim text from file
type f(type,...); optional function declaration
type name; variable declaration
type f(type arg,...) {} function definition

Data types/declarations

boolean (true or false) bool
tri-state boolean (true, default, or false) bool3
integer int
float (double precision) real
ordered pair (complex number) pair
color, line type/width/cap, font, fill rule guide
label with position, alignment, pen attributes Label
drawing canvas picture
affine transform transform
constant (unchanging) value const
allocate in higher scope static
no value void
inhibit implicit argument casting explicit
structure typedef type name

3D data types (import three)

ordered triple triple
3D path path3
3D guide guide3
3D affine transform transform3

Constants

exponential form 6.02e23
\TeX string constant "abc\de"
\TeX strings: special characters \\, \"\n\c, \r, \t, \b
C strings: constant 'abc\de'
C strings: special characters \\, \"\n\c, \r, \t, \b
C strings: newline, cr, tab, backspace \n \r \t \b
C strings: octal, hexadecimal bytes \0-\377 \x0-\x7F

Operators

arithmetic operations + - * /
modulo (remainder) %
comparisons == != > >= < <=
not !
and or (conditional evaluation of RHS) && ||
and or xor & | ^
cast expression to type ((type) expr)
increment decrement prefix operators ++ --
constant expression expr1 ? expr2 : expr3
assignment operators name.member
conditional expression expression evaluation separator,

Flow control

statement terminator ;
block delimiters {
comment delimiters /* */
comment to end of line delimiter //
exit from while/while/for break;
next iteration of while/while/for continue;
return value from function return expr;
terminate execution exit();
abort execution with error message abort(string);

Flow constructions (if/while/for/do)

if(expr) statement
else if(expr) statement
else statement
while(expr) statement
for(expr1; expr2; expr3) statement
for(type var : array) statement
do statement
while(expr);
### Arrays

```plaintext
array
array element i
array indexed by elements of int array A
anonymous array
array containing n deep copies of x length
cyclic flag
pop element x
push element x
append array a
insert rest arguments at index i
delete element at index i
delete elements with indices in [1,j]
delete all elements
test whether element n is initialized
array of indices of initialized elements
complement of int array in {0,...,n-1}
deep copy of array a
array {0,1,...,n-1} array {n,n+1,...,m}
array {n-1,n-2,...,0} array {f(0),f(1),...,f(n-1)}
array obtained by applying f to array a
uniform partition of [a,b] into n intervals
concat specified 1D arrays
return sorted array
return array sorted using ordering
push element x
pop element x
cyclic flag
name
length
cyclic
pop()
push(x)
append(a)
sort(i,1,...)
delete(i)
delete(i,j)
delete()
initialized(n)
keys
length
initialized
name
```

### Initialization

```plaintext
initial variable
initialize array
```

### path connectors

```plaintext
straight segment
Bezier segment with implicit control points
Bezier segment with explicit control points
concatenate
lift pen
```

### Labels

```plaintext
implicit cast of string s to Label
Label a with relative position and alignment
Label a with absolute position and alignment
Label a with specified pen
```

### draw commands

```plaintext
draw path with current pen
draw path with pen
draw labeled path
draw arrow with pen
draw path on picture
draw visible portion of line through two pairs
```

### fill commands

```plaintext
fill path with current pen
fill path with pen
fill path on picture
```

### label commands

```plaintext
label a pair with optional alignment z
label a path with optional alignment z
add label to picture
```

### clip commands

```plaintext
clip to path
clip to path with fill rule
clip picture to path
```

### pens

```plaintext
Grayscale pen from value in [0,1]
RGB pen from values in [0,1]
CMYK pen from values in [0,1]
RGB pen from heximdecimal string
heximdecimal string from rgb pen
hsb pen from values in [0,1]
invisible pen
default pen
current pen
solid pen
dotted pen
wide dotted current pen
wide dotted pen
```

### fill commands

```plaintext
fill(path)
fill(path,pen)
fill(picture,path)
```

### label commands

```plaintext
label(Label,pair,z)
label(label,path,z)
label(label,path,Label)
```

### clip commands

```plaintext
clip(path)
clip(path,pen)
clip(picture,path)
```

### pens

```plaintext
gray(g)
rgb(r,g,b)
cmyk(r,g,b)
rgb(string)
hsb(pen)
invisible
defaultpen
currentpen
solid
dotted
Dotted
Dotted(pen)
dashed
longdashed
dashdotdotted
squaredotted
currentpen
defaultpen
invisible
```

### fill commands

```plaintext
fill(path)
fill(path,pen)
fill(picture,path)
```
path operations

number of segments in path \( p \)
number of nodes in path \( p \)
is segment \( i \) of path \( p \) straight?
is path \( p \) cyclic?
coordinates of path \( p \) at time \( t \)
direction of path \( p \) at time \( t \)
direction of path \( p \) at length \( (p) \)
unit \( \text{dir}(p)+\text{dir}(q) \)
acceleration of path \( p \) at time \( t \)
radius of curvature of path \( p \) at time \( t \)
precontrol point of path \( p \) at time \( t \)
postcontrol point of path \( p \) at time \( t \)
arclength of path \( p \)
time at which \( \text{arclength}(p)=L \)
point on path \( p \) at arclength \( L \)
first value \( t \) at which \( \text{dir}(p,t)=z \)
time \( t \) at relative fraction \( t \) of \( \text{arclength}(p) \)
point midway along arclength of \( p \)
path running backwards along \( p \)
subpath of \( p \) between times \( a \) and \( b \)
times for one intersection of paths \( p \) and \( q \)
times at which \( p \) reaches minimal extents
times at which \( p \) reaches maximal extents
intersection times of paths \( p \) and \( q \)
intersection times of path \( p \) with \( "-a--b--c" \)
intersection times of path \( p \) crossing \( x=x \)
intersection times of path \( p \) crossing \( y=y \)
intersection point of paths \( p \) and \( q \)
intersection points of \( p \) and \( q \)
intersection of extension of \( P--Q \) and \( p--q \)
lower left point of bounding box of path \( p \)
upper right point of bounding box of path \( p \)
subpaths of \( p \) split by \( n \)th cut of \( \text{knife} \)
pair \( z \) lies within path \( p \)
pair \( z \) lies within or on path \( p \)
path surrounding region bounded by paths \( \text{path filled by } \text{draw}(g,p) \)
unit square with lower-left vertex at origin
unit circle centered at origin
circle of radius \( r \) about \( c \)
arc of radius \( r \) about \( c \) from angle \( a \) to \( b \)
unit \( n \)-sided polygon
unit \( n \)-point cyclic cross
pictures

add picture \( \text{pic} \) to \( \text{currentpicture} \)
add picture \( \text{pic} \) about pair \( z \)

affine transforms

identity transform
shift by values
shift by pair
scale by \( x \) in the \( x \) direction
scale by \( y \) in the \( y \) direction
scale by \( x \) in both directions
scale by real values \( x \) and \( y \)
map \((x,y)\rightarrow(x+ay,y)\)
rotate by real angle in degrees about pair \( z \)
reflect about line from \( P--Q \)

string operations

concatenation operator
string length
position \( \geq \) pos of first occurrence of \( t \) in \( s \)
position \( \leq \) pos of last occurrence of \( t \) in \( s \)
string \( s \) with \( n \) characters at pos erased
substring of string \( s \) of length \( n \) at pos
string \( s \) reversed
string \( s \) with before changed to after
string \( s \) translated via \( \{\{\text{before},\text{after}\}\}\)
format \( x \) using C-style format string \( s \)
casts hexadecimal string to an integer
casts \( x \) to string using precision digits
current time formatted by format
time in seconds of string \( t \) using format
string corresponding to seconds using format
split \( s \) into strings separated by \( \text{delimiter} \)

pictures

add picture \( \text{pic} \) to \( \text{currentpicture} \)
add picture \( \text{pic} \) about pair \( z \)

May 2014 v1.1. Copyright © 2014 John C. Bowman
Permission is granted to make and distribute copies of this card, with or without modifications, provided the copyright notice and this permission notice are preserved on all copies.