

Announcements

Thursday Extra 4:00 in CS Commons/Science 3821
none this week

Schedule this week a variation of last week

Class as normal Mon., Wed.; Project all day Fri.

Office hours:

Mon., Wed.: 1:30-3:00 pm

Tues.: 2:30 - 4:00 pm

Thurs.: uncertain

Fri.: canceled

Mentor sessions Tues. 6-7 and 7-8 (pm) Commons

Functions with parameters

Questions

Clicker Questions

Pre-conditions, Testing

Questions

Clicker Questions

What is printed by the following program:

```
#include <stdio.h>

void procA (int r, int * s)
{ printf ("procA-1: r: %2d, *s: %2d\n", r, *s);
  r = 12;
  *s = 17;
  printf ("procA-2: r: %2d, *s: %2d\n", r, *s);
}

void procB (int x, int * y)
{ printf ("procB-1: x: %2d, *y: %2d\n", x, *y);
  x = 22;
  *y = 29;
  procA (*y, &x);
  printf ("procB-2: x: %2d, *y: %2d\n", x, *y);
}

int main()
{ int a = 4;
  int b = 8;
  printf ("main-1: a: %2d, a: %2d\n", a, b);
  procB(a, &b);
  printf ("main-2: a: %2d, a: %2d\n", a, b);
  return 0;
}
```

Consider the program

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

```
double addThreeSides//add 2 sides+bottom of rectangle  
    (double width, double length)
```

```
{  
    double sum = 2.0 * width+ length;  
    return sum;  
}  
int main ()  
{  
    double width = 3.0;  
    double length = 5.0;  
    addThreeSides;  
    printf ("a %.1lf by %.1lf rectangle\n",  
        width, length);  
    printf ("has sum %.1lf\n",  
        sum);  
    return 0;  
}
```

What happens if we try to compile and run the program?

- A. program does not compile or does not run
- B. a 3.0 by 5.0 rectangle has sum ?????
- C. a 3.0 by 5.0 rectangle has sum 11.0
- D. a 3.0 by 5.0 rectangle has sum 13.0
- E. a 5.0 by 3.0 rectangle has sum 13.0

Consider the program

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

```
double addThreeSides//add 2 sides+bottom of rectangle  
    (double width, double length)
```

```
{  
    double sum = 2.0 * width+ length;  
    return sum;  
}  
int main ()  
{  
    double width = 3.0;  
    double length = 5.0;  
    addThreeSides (width, length);  
    printf ("a %.1lf by %.1lf rectangle\n",  
        width, length);  
    printf ("has sum %.1lf\n",  
        sum);  
    return 0;  
}
```

What happens if we try to compile and run the program?

- A. program does not compile or does not run
- B. a 3.0 by 5.0 rectangle has sum ?????
- C. a 3.0 by 5.0 rectangle has sum 11.0
- D. a 3.0 by 5.0 rectangle has sum 13.0
- E. a 5.0 by 3.0 rectangle has sum 13.0

Consider the program

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

```
double addThreeSides //add 2 sides+bottom of rectangle
```

```
(double width, double length)
```

```
{ double sum = 2.0 * width+ length;
```

```
  return sum;
```

```
}
```

```
int main ()
```

```
{
```

```
  double width = 3.0;
```

```
  double length = 5.0;
```

```
  double sum;
```

```
  sum = addThreeSides (width, length);
```

```
  printf ("a %.1lf by %.1lf rectangle\n",
```

```
    width, length);
```

```
  printf ("has sum %.1lf\n",
```

```
    sum);
```

```
  return 0;
```

```
}
```

What happens if we try to compile and run the program?

- A. program does not compile or does not run
- B. a 3.0 by 5.0 rectangle has sum ?????
- C. a 3.0 by 5.0 rectangle has sum 11.0
- D. a 3.0 by 5.0 rectangle has sum 13.0
- E. a 5.0 by 3.0 rectangle has sum 13.0

Consider the program

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

```
double addThreeSides
```

```
    (double width, double length)
```

```
{
```

```
    double sum = 2.0 * width+ length;
```

```
    return sum;
```

```
}
```

```
int main ()
```

```
{
```

```
    double width = 3.0;
```

```
    double length = 5.0;
```

```
    double sum;
```

```
    sum
```

```
        = addThreeSides (length, width);
```

```
    printf ("a %.1lf by %.1lf rectangle\n",
```

```
           width, length);
```

```
    printf ("has sum %.1lf\n",
```

```
           sum);
```

```
    return 0;
```

```
}
```

What happens if we try to compile and run the program?

- A. program does not compile or does not run
- B. a 3.0 by 5.0 rectangle has sum ?????
- C. a 3.0 by 5.0 rectangle has sum 11.0
- D. a 3.0 by 5.0 rectangle has sum 13.0
- E. a 5.0 by 3.0 rectangle has sum 13.0

Consider the following C program.

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

```
int main ()
```

```
{ proc (7);
```

```
    return 0;
```

```
}
```

```
void proc (double dbl)
```

```
{ printf ("in prod, dbl is %lf\n", dbl);
```

```
}
```

What happens if we try to compile and run this program?

1. Compilation error

2. Run-time error

3. prints: in prod, dbl is 7

4. prints: in prod, dbl is #%@ (some apparently random number)

5. something else

```
/* function to compute the larger  
   root for  $ax^2 + bx + c = 0$  */  
double findRoot (double a, double b, double c)  
{  
    double disc =  $b*b - 4*a*c$ ;  
    return  $(-b + \text{sqrt}(\text{disc})) / (2 * a)$ ;  
}
```

Some possible pre-conditions for this function might be:

1. $b*b - 4*a*c \geq 0$
2. $a \neq 0$
3. a, b, and c are doubles

Which of these should be stated explicitly for this function?

- | | |
|---|-------------|
| A. 1 and 2 | C. 2, not 1 |
| B. 1, not 2 | D. neither |
| E. I'm not sure how to subtract "conditions" from "pre" | |

```
/* function to compute the larger  
   root for  $ax^2 + bx + c = 0$  */  
double findRoot (double a, double b, double c)  
{  
    double disc =  $b*b - 4*a*c$ ;  
    return  $(-b + \text{sqrt}(\text{disc})) / (2 * a)$ ;  
}
```

Some possible pre-conditions for this function might be:

1. $b*b - 4*a*c \geq 0$
2. $a \neq 0$
3. a, b, and c are doubles

Should condition 3 be stated explicitly for this function?

- A. Yes
- B. No

Consider the following two statements:

1. Suppose a function header specifies: Pre-conditions: None and the program compiles, then processing should work properly for any data supplied.
2. Suppose a function header specifies: Pre-conditions: None and the program compiles, then processing may proceed in any way whatsoever.

Which, if any, of these statements is valid?

- | | |
|-------------|-------------|
| A. 1 and 2 | C. 2, not 1 |
| B. 1, not 2 | D. neither |

Consider the following two statements:

1. Suppose a function header specifies: Post-conditions: None and the program compiles, then processing may proceed in any way whatsoever.
2. Suppose a function header specifies: Post-conditions: None and the program compiles, then the program would work properly if the program crashed during processing.

Which, if any, of these statements is valid?

- | | |
|-------------|-------------|
| A. 1 and 2 | C. 2, not 1 |
| B. 1, not 2 | D. neither |

Supplemental Problem 1 specified starting and stopping times in three ranges

- 7:00 am - 5:00 pm
- 5:00 pm - midnight
- midnight - 7:00 am

Suppose also that the program subdivides the daytime range into two pieces: 7:00 am - noon and noon - 5:00 pm

Ignoring error checking, about how many tests should be considered for White Box Testing?

- A. 1-5
- B. 6-9
- C. 10-13
- D. 14-16
- E. more than 16

Supplemental Problem 1 specified starting and stopping times in three ranges

- 7:00 am - 5:00 pm
- 5:00 pm - midnight
- midnight - 7:00 am

Suppose also that the program subdivides the daytime range into two pieces: 7:00 am - noon and noon - 5:00 pm

Ignoring error checking, about how many tests should be considered for Black Box Testing?

- A. 1-5
- B. 6-9
- C. 10-13
- D. 14-16
- E. more than 16

Lab today

- several optional topics
 - functions as parameters
 - arrays of functions
 - work with get-ir.c
- focus on non-optional parts
 - pre- and post-conditions
 - assert statement
 - choosing test cases
 - debugging
 - practice with parameters
 - indicating function success
- work through optional sections as time and interest permit
- optional sections are not integral to the course

Since today's lab largely continues work last week,

- same lab partners today as last week
- new lab partners assigned Wednesday