VE280 Recitation Class (2)

Prepared by,
Zhang Yuhang
Yu JinZe
Cheng Songzhe
Yan Xuebin

Data Structure Summer
May 31st, 2012
E-Building, R2-103

Outline

- About Project One
- Review Linux command
- Makefile
- Review of C++
  - Variable
  - Operator
  - Flow-control
  - Function

Project 1

- use forum

Review of Linux command

- ls cd
- path
- g++ -Wall –Werror –o p1 io.C main.C
  ./p1 < test > testout
  diff testout test-result

GDB revisit

- General procedure:
  - compile with $g++ -g xxx.c xxx.out
  - $gdb xxx.out
  - (gdb)run
    - To see if there's any bugs or errors
  - (gdb)break
    - insert breakpoints
  - (gdb)list
    - list codes

Makefile

- p1: main.C

  g++ -o p1 main.C

- all:p1

  p1: main.o io.o

  g++ -o p1 main.o io.o

  main.o:main.C

  g++ -c main.c

  io.o:io.C

  g++ -c io.C

Makefile

Macro Def

Dasha=“shuai”

Main:

```
	echo $Dasha
```

p1:p1.C

```
	g++ -o $@ $<
```

$@ p1

$< p1.C

$(MAINSRCS:.C=)

p1.C ➔ p1

Makefile

```
CC = g++
MAINSRCS = multi_source.C
OTHSRCS = say_hello.C
CFLAGS = -g -Wall
SRCS = $(MAINSRCS) $(OTHSRCS)
OBJS = $(SRCS:.C=.o)
TARGETS = $(MAINSRCS:.C=)

%.o: %.C
	$(CC) $(CFLAGS) -o $@ -c $<
```

Makefile

```
CC = g++
MAINSRCS = multi_source.C
OTHSRCS = say_hello.C
CFLAGS = -g -Wall
SRCS = $(MAINSRCS) $(OTHSRCS)
OBJS = $(SRCS:.C=.o)
TARGETS = $(MAINSRCS:.C=)

%.o: %.C
	$(CC) $(CFLAGS) -o $@ -c $<

all: $(TARGETS)

$(TARGETS): $(OBJS)
	$(CC) $(CFLAGS) -o $(TARGETS) $(OBJS)

clean:
	rm -f $(OBJS) $(TARGETS)
```

Variable

- Built-in Types

  int, double, float, char, string …

  usigned, long, short …

typedef

- Cast

  Int a;

  Double b;

  a = (int)b;

  a = int(b);

  a = static_cast<int>(b);

Loop Control

- if-else
- switch
- for
- while & do-while
- break and continue

If-Else

- if-else
  - if (stmt) {block} [else if (stmt) {block}] [else {block}]
  - example:

```
if (myScore > yourScore)
        cout << “I win!\n”;
else if (myScore < yourScore)
        cout << “You are so lucky!”;
else //may check if there’s unexpected situations
        cout << “Did you copy my paper?”;
```

switch

- switch

```
switch (vehicleClass)
{
    case 1:
        cout << "Passenger car.";
        toll = 0.50;
        break;
    case 2:
        cout << "Bus.";
        toll = 1.50;
        break;
    case 3:
        cout << "Truck.";
        toll = 2.00;
        break;
    default: //optional, but can check unexpected conditions
        cout << "Unknown vehicle class!";
}
```

for & while

- for

```
for(int i = 0; i < 10; i++)
    cout << “This is the “ << i+1 << “times”;
```

- while

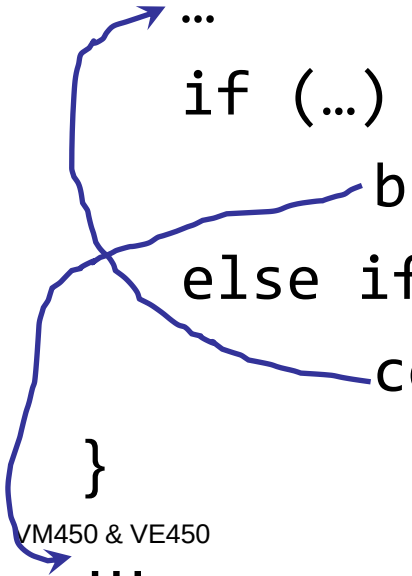
```
int countDown = 5;
while(--countDown >= 0)
    cout << countDown << “!\n”;
cout << “BOOM!!!”
```

this prgrm’s output??

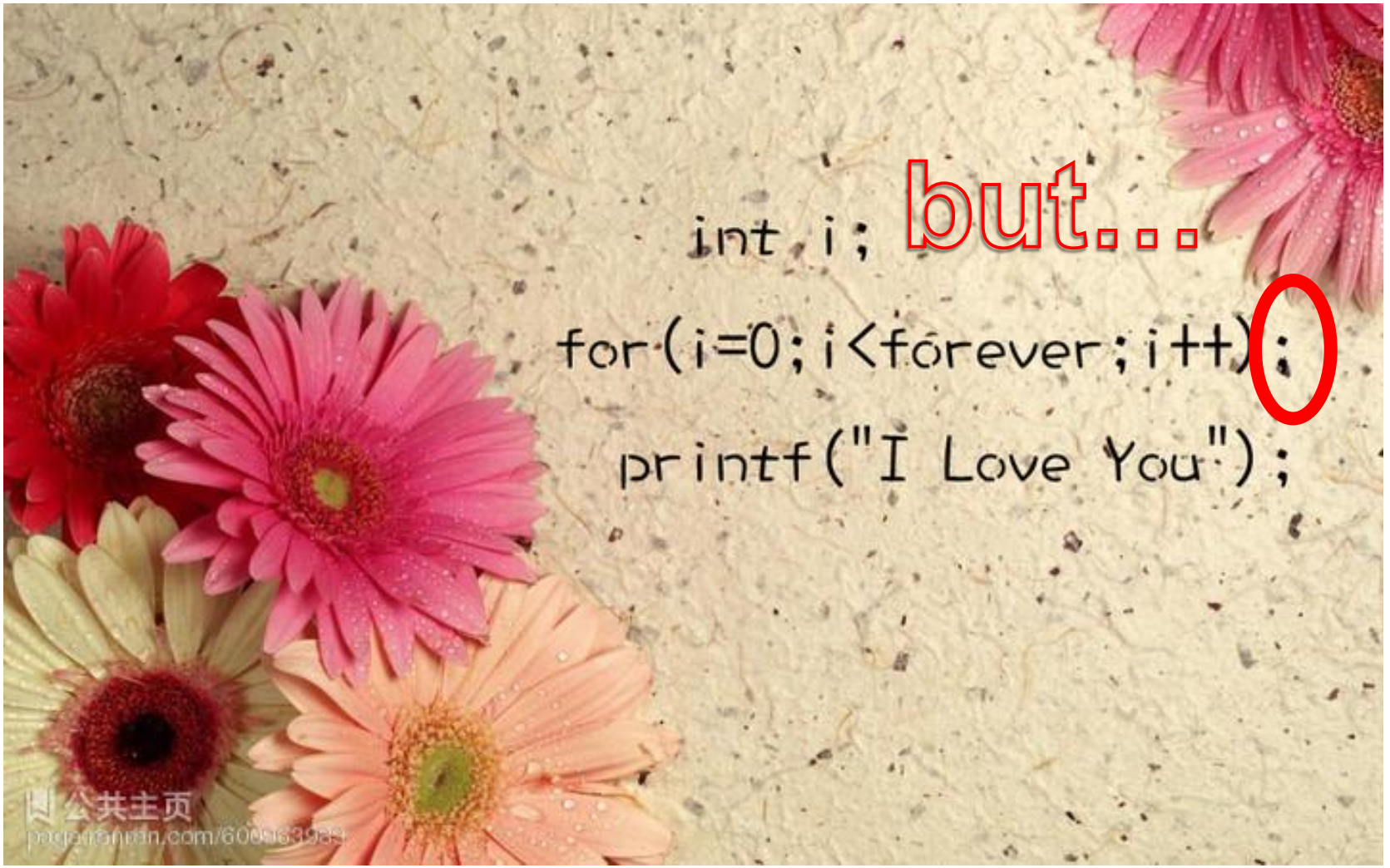
Break and Continue

- break
  - jump out of the loop
- continue
  - ends the current iteration of the loop body

```
while(num > 0) {
    …
    if (…)
        break;
    else if (…)
        continue;
}
```



Beautiful at first sight!



functions

- scope
  - notice scope of for:

```
for (int i = 0; i < 10; i++)
        cout << i << endl;
cout << i;
```

- default parameters