

# Summary of Lesson 13

**ex1.c**

```
#include "ext.h"

// zl group

typedef struct {
    t_object b_ob;
    void *listout;
    t_atom *array;
    long listcount;
} t_ex1;

void ex1_bang(t_ex1 *x);
void ex1_int(t_ex1 *x, long input);

void *ex1_new(void);
void ex1_free(t_ex1 *x);

void *ex1_class;

void main()
{
    setup((t_messlist **)&ex1_class, (method)ex1_new, (method)ex1_free,
    (short)sizeof(t_ex1), 0L, 0L);
    addbang((method) ex1_bang);
    addint((method) ex1_int);
}

void ex1_bang(t_ex1 *x)
{
    outlet_list(x->listout, 0L, x->listcount, x->array);
    x->listcount = 0;
}

void ex1_int(t_ex1 *x, long input)
{
    SETLONG(x->array+x->listcount,input);

    //x->array[x->listcount].a_type = A_LONG;
    //x->array[x->listcount].a_w.w_long = input;
    x->listcount++;
}

void *ex1_new(void)
{
    t_ex1 *x;
```

```

    x = (t_ex1*)newobject(ex1_class);
    x->array = (t_atom*)malloc(sizeof(t_atom) * 100);
    x->listout = listout(x);
    x->listcount = 0;
    return x;
}

void ex1_free(t_ex1 *x)
{
    if(x->array)
        free(x->array);
}

```

---

## ex2.c

```

#include "ext.h"

// hash table

typedef struct {
    t_object b_ob;
    void *out;
    t_atom *array;
    long count;
} t_ex2;

void ex2_bang(t_ex2 *x);
void ex2_int(t_ex2 *x, long input);

void *ex2_new(void);
void ex2_free(t_ex2 *x);

void *ex2_class;

void main()
{
    setup((t_messlist **)&ex2_class, (method)ex2_new, (method)ex2_free,
    (short)sizeof(t_ex2), 0L, 0L);
    addbang((method) ex2_bang);
    addint((method) ex2_int);
}

void ex2_bang(t_ex2 *x)
{
    outlet_anything(x->out, gensym("myMessage"), x->count, x->array);
    x->count = 0;
}

void ex2_int(t_ex2 *x, long input)
{

```

```

        SETLONG(x->array+x->count,input);
        x->count++;
    }

void *ex2_new(void)
{
    t_ex2 *x;
    x = (t_ex2*)newobject(ex2_class);
    x->array = (t_atom*)malloc(sizeof(t_atom) * 100);
    x->out = outlet_new(x, 0L);
    x->count = 0;
    return x;
}

void ex2_free(t_ex2 *x)
{
    if(x->array)
        free(x->array);
}

```

---

### ex3.c

```

#include "ext.h"

// funnel - > make real funnel

typedef struct {
    t_object b_ob;
    void *listout;
    long id[5];
    void *proxy;
} t_ex3;

void ex3_bang(t_ex3 *x);
void ex3_int(t_ex3 *x, long input);

void *ex3_new(void);
void ex3_free(t_ex3 *x);

void *ex3_class;

void main()
{
    setup((t_messlist **)&ex3_class, (method)ex3_new, (method)ex3_free,
    (short)sizeof(t_ex3), 0L, 0L);
    addint((method) ex3_int);
}

void ex3_int(t_ex3 *x, long input)
{

```

```

    t_atom list[2];
    SETLONG(&list[0] , proxy_getinlet((t_object*)x));
    SETLONG(&list[1] , input);
    outlet_list(x->listout, 0L, 2, list);
}

void *ex3_new(void)
{
    int i;
    t_ex3 *x;
    x = (t_ex3*)newobject(ex3_class);
    x->listout = listout(x);
    for(i = 5;i;i--)
    {
        x->proxy = proxy_new(x,i,&x->id[i]);
    }
    return x;
}

void ex3_free(t_ex3 *x)
{
    if(x->proxy)
        free(x->proxy);
}

```

---

#### **ex4.c**

```

#include "ext.h"

// assignment make a switch which accept only int

typedef struct {
    t_object b_ob;
    void *out;
    long *id;
    int which;
    void *proxy;
} t_ex4;

void ex4_bang(t_ex4 *x);
void ex4_int(t_ex4 *x, long input);

void *ex4_new(long numinlet);
void ex4_free(t_ex4 *x);

void *ex4_class;

void main()
{

```

```

        setup((t_messlist **)&ex4_class, (method)ex4_new, (method)ex4_free,
(short)sizeof(t_ex4), 0L, A_LONG, 0L);
        addint((method) ex4_int);
    }

void ex4_int(t_ex4 *x, long input)
{
    int inlet;
    inlet = proxy_getinlet((t_object*)x);
    if(!inlet)
    {
        x->which = input;
    }
    if(inlet == x->which)
    {
        outlet_int(x->out,input);
    }
}

void *ex4_new(long numinlet)
{
    int i;
    t_ex4 *x;
    x = (t_ex4*)newobject(ex4_class);
    x->out = intout(x);
    x->id = (long*)malloc(sizeof(long) * numinlet);
    for(i = numinlet;i;i--)
    {
        x->proxy = proxy_new(x,i,x->id+i);
    }
    return x;
}

void ex4_free(t_ex4 *x)
{
    if(x->proxy)
        free(x->proxy);

    if(x->id)
        free(x->id);
}

```

---

## ex5.c

```
#include "ext.h"
```

```
typedef struct {
    t_object b_ob;
```

```

    void* clock;
    void *out;
} t_ex5;

void ex5_tick(t_ex5 *x);
void ex5_int(t_ex5 *x, long input);

void *ex5_new(void);
void ex5_free(t_ex5 *x);

void *ex5_class;

void main()
{
    setup((t_messlist **)&ex5_class, (method)ex5_new, (method)ex5_free,
(short)sizeof(t_ex5), 0L, 0L);
    addint((method) ex5_int);
}

void ex5_int(t_ex5 *x, long input)
{
    clock_delay(x->clock, input);
}

void ex5_tick(t_ex5 *x)
{
    outlet_bang(x->out);
}

void *ex5_new(void)
{
    t_ex5 *x;
    x = (t_ex5*)newobject(ex5_class);
    x->out = bangout(x);
    x->clock = clock_new(x,(method)ex5_tick);
    return x;
}

void ex5_free(t_ex5 *x)
{
    freeobject(x->clock);
}

```

---

## ex6.c

```
#include "ext.h"
```

```
// with stop
```

```
typedef struct {
```

```

    t_object b_ob;
    void* clock;
    void *out;
} t_ex6;

void ex6_tick(t_ex6 *x);
void ex6_int(t_ex6 *x, long input);
void ex6_stop(t_ex6 *x);

void *ex6_new(void);
void ex6_free(t_ex6 *x);

void *ex6_class;

void main()
{
    setup((t_messlist **)&ex6_class, (method)ex6_new, (method)ex6_free,
(short)sizeof(t_ex6), 0L, 0L);
    addint((method) ex6_int);
    addmess((method) ex6_stop, "stop", 0L);
}

void ex6_int(t_ex6 *x, long input)
{
    clock_delay(x->clock, input);
}

void ex6_stop(t_ex6 *x)
{
    clock_unset(x->clock);
}

void ex6_tick(t_ex6 *x)
{
    outlet_bang(x->out);
}

void *ex6_new(void)
{
    t_ex6 *x;
    x = (t_ex6*)newobject(ex6_class);
    x->out = bangout(x);
    x->clock = clock_new(x,(method)ex6_tick);
    return x;
}

void ex6_free(t_ex6 *x)
{
    freeobject(x->clock);
}

```

---

**ex7.c**

```
#include "ext.h"

// get logical time

typedef struct {
    t_object b_ob;
    void* clock;
    void *out;
} t_ex7;

void ex7_bang(t_ex7 *x);

void *ex7_new(void);
void ex7_free(t_ex7 *x);

void *ex7_class;

void main()
{
    setup((t_messlist **)&ex7_class, (method)ex7_new, 0L, (short)sizeof(t_ex7),
    0L, 0L);
    addbang((method) ex7_bang);
}

void ex7_bang(t_ex7 *x)
{
    outlet_int(x->out, gettimeofday());
    // outlet_int(x->out, systime_ticks());
}

void *ex7_new(void)
{
    t_ex7 *x;
    x = (t_ex7*)newobject(ex7_class);
    x->out = intout(x);
    return x;
}
```

---

**ex8.c**

```
#include "ext.h"

// get date time

typedef struct {
    t_object b_ob;
```



```

    void* clock;
} t_ex8;

void ex8_bang(t_ex8 *x);

void *ex8_new(void);
void ex8_free(t_ex8 *x);

void *ex8_class;

void main()
{
    setup((t_messlist **)&ex8_class, (method)ex8_new, 0L, (short)sizeof(t_ex8),
    0L, 0L);
    addbang((method) ex8_bang);
}

void ex8_bang(t_ex8 *x)
{
    t_datetime datetime;
    systime_datetime(&datetime);
    post("%d/%d/%d %d:%d:%d",datetime.day, datetime.month,datetime.year,
datetime.hour, datetime.minute, datetime.second);
}

void *ex8_new(void)
{
    t_ex8 *x;
    x = (t_ex8*)newobject(ex8_class);
    return x;
}

```