

# Summary of Lesson 11

## Functions , introduced in this lesson

```
// attribute of an object
void setup
(t_messlist **class, method createfun, method freefun, short classSize, method
menufun, short types...);

//for debug
void post (char *fmtstring, void *items...);

// add an alias name
void alias (char *name);
// define with defferent name
void class_setname (char *obname, char *filename);

// for defining callback function for leftmost inlet
void addbang (method mp);
void addfloat (method mp);
void addint (method mp);
void addmess (method mp; char *sym; short types...);
```

---

### ex1.c

```
// setup
#include "ext.h"

typedef struct {
    t_object b_ob;
} t_ex1;

void ex1_bang(t_ex1 *x);
void *ex1_new(void);
void ex1_free(void);

void *ex1_class;

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

void *ex1_new(void)
{
    t_ex1 *x;
    x = (t_ex1*)newobject(ex1_class);
```

```

    post("I am here in the max patch!");
    return x;
}

void ex1_free(void)
{
    post("Jesus! Are you gonna kill me?");
}

```

---

## ex2.c

```

//setup with arguments
#include "ext.h"

typedef struct {
    t_object b_ob;
} t_ex2;

void ex2_bang(t_ex2 *x);
void *ex2_new(long a, long b, long c, long d);
void ex2_free(void);

void *ex2_class;

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

void *ex2_new(long a, long b, long c, long d)
{
    t_ex2 *x;
    x = (t_ex2*)newobject(ex2_class);
    post("sum of arguments = %d", a + b + c + d);
    return x;
}

void ex2_free(void)
{
    post("So what?");
}

```

---

## ex3.c

```

#include "ext.h"
//setup with arguments, including "DEF"

typedef struct {
    t_object b_ob;
}

```

```

} t_ex3;

void *ex3_new(long a, float b);

void *ex3_class;

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

void *ex3_new(long a, float b)
{
    t_ex3 *x;
    x = (t_ex3*)newobject(ex3_class);

    post("arguments %d %f", a , b);

    return x;
}

```

---

#### ex4.c

```

#include "ext.h"
// set alias
// must be installed in the start-up folder

typedef struct {
    t_object b_ob;
} t_ex4;

void *ex4_new(long a, float b);
void ex4_free(void);

void *ex4_class;

void main()
{
    setup((t_messlist **)&ex4_class, (method)ex4_new, 0L, (short)sizeof(t_ex4),
    0L, A_LONG, A_FLOAT, 0L);
    alias("Beispiel4");
}

void *ex4_new(long a, float b)
{
    t_ex4 *x;
    x = (t_ex4*)newobject(ex4_class);
    return x;
}

```

```
}
```

---

### **ex5.c**

```
#include "ext.h"
// set different name from file name
// must be installed in the start-up folder

typedef struct {
    t_object b_ob;
} t_ex5;

void *ex5_new(long a, float b);
void *ex5_class;

void main()
{
    setup((t_messlist **)&ex5_class, (method)ex5_new, 0L, (short)sizeof(t_ex5),
    0L, A_LONG, A_FLOAT, 0L);
    class_setname("../", "ex5");
}

void *ex5_new(long a, float b)
{
    t_ex5 *x;
    x = (t_ex5*)newobject(ex5_class);
    return x;
}
```

---

### **ex6.c**

```
#include "ext.h"
// addbang

typedef struct {
    t_object b_ob;
} t_ex6;

void ex6_bang(void);
void *ex6_new(void);
void *ex6_class;

void main()
{
    setup((t_messlist **)&ex6_class, (method)ex6_new, 0L, (short)sizeof(t_ex6),
    0L, 0L);
    addbang((method)ex6_bang);
}
```

```

void *ex6_new(void)
{
    t_ex6 *x;
    x = (t_ex6*)newobject(ex6_class);
    return x;
}

void ex6_bang(void)
{
    post("Jesus! Are you gonna kill me?");
}

```

---

### **ex7.c**

```

#include "ext.h"
// addint

typedef struct {
    t_object b_ob;
} t_ex7;

void ex7_int(t_ex7* x, int value);
void *ex7_new(void);
void *ex7_class;

void main()
{
    setup((t_messlist **)&ex7_class, (method)ex7_new, 0L, (short)sizeof(t_ex7),
    0L, 0L);
    addint((method)ex7_int);
}

void *ex7_new(void)
{
    t_ex7 *x;
    x = (t_ex7*)newobject(ex7_class);
    return x;
}

void ex7_int(t_ex7* x, int value)
{
    post("I got %d",value);
}

```

---

### **ex8.c**

```

#include "ext.h"
// addfloat

typedef struct {

```

```

    t_object b_ob;
} t_ex8;

void ex8_float(t_ex8* x, float value);
void *ex8_new(void);
void *ex8_class;

void main()
{
    setup((t_messlist **)&ex8_class, (method)ex8_new, 0L, (short)sizeof(t_ex8),
0L, 0L);
    addfloat((method)ex8_float);
}

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

void ex8_float(t_ex8* x, float value)
{
    post("I got %f",value);
}

```

---

## ex9.c

```

#include "ext.h"
// address

typedef struct {
    t_object b_ob;
} t_ex9;

void ex9_greetingA(t_ex9 *x);
void ex9_greetingB(t_ex9 *x);

void *ex9_new(long a, float b);

void *ex9_class;

void main()
{
    setup((t_messlist **)&ex9_class, (method)ex9_new, 0L, (short)sizeof(t_ex9),
0L, 0L);
    address((method)ex9_greetingA , "Hello", 0L);
    address((method)ex9_greetingB , "Bye", 0L);
}

```

```

void *ex9_new(long a, float b)
{
    t_ex9 *x;
    x = (t_ex9*)newobject(ex9_class);
    return x;
}

void ex9_greetingA(t_ex9 *x)
{
    post("Hello, How are you?");
}

void ex9_greetingB(t_ex9 *x)
{
    post("Good bye, Have a nice day");
}

```

---

## ex10.c

```

#include "ext.h"
// more primitive than "i" object

typedef struct {
    t_object b_ob;
    int val; // value for sharing
} t_ex10;

void ex10_bang(t_ex10* x);
void ex10_int(t_ex10* x, int value);
void *ex10_new(void);
void *ex10_class;

void main()
{
    setup((t_messlist **)&ex10_class, (method)ex10_new, 0L,
    (short)sizeof(t_ex10), 0L, 0L);
    addint((method)ex10_int);
    addbang((method)ex10_bang);
}

void *ex10_new(void)
{
    t_ex10 *x;
    x = (t_ex10*)newobject(ex10_class);
    x->val = 0;
    return x;
}

void ex10_bang(t_ex10* x)
{
    post("%d",x->val);
}

```

```
}
```

```
void ex10_int(t_ex10* x, int value)
```

```
{
```

```
    x->val = value;
```

```
}
```