AN OVERVIEW OF C

System Programming Basic Exercise's lecture note 3

Remind

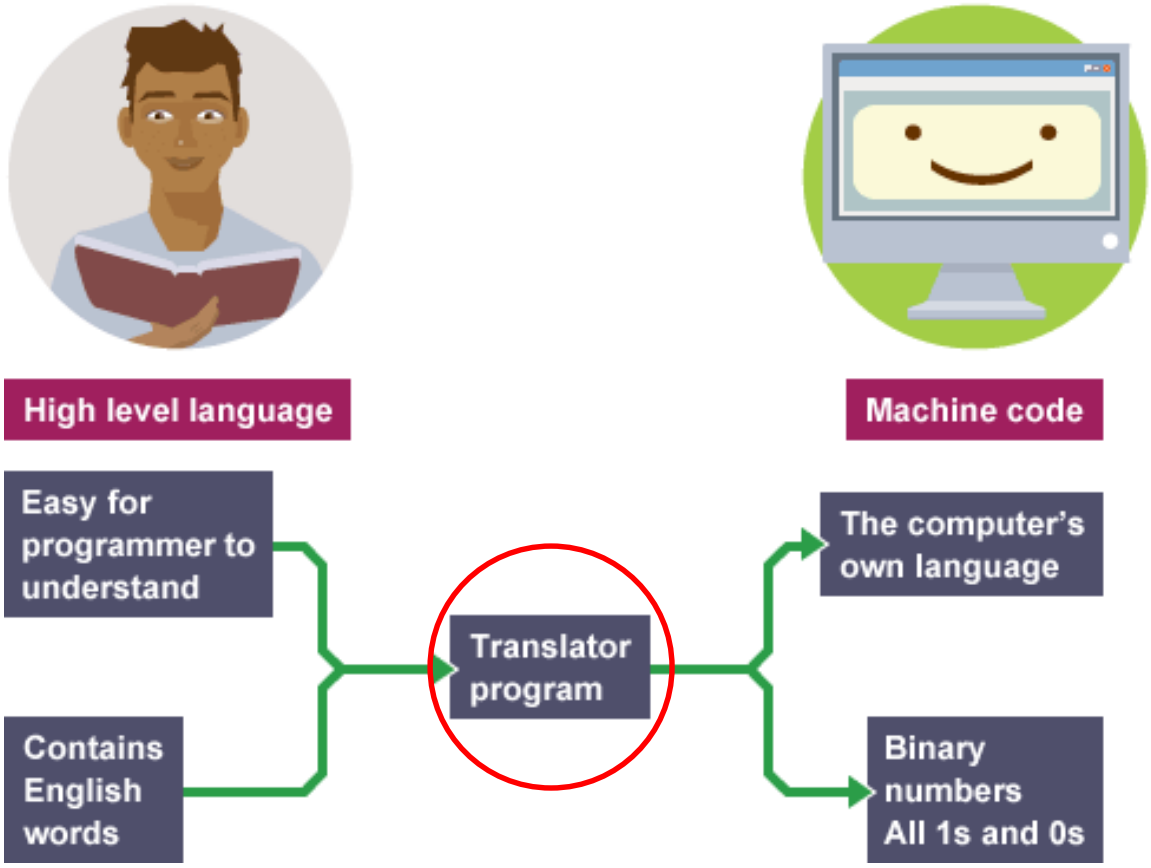
- C언어 - Hello World!

```
$ gcc hello.c –o hello
$ ./hello
```

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
  printf("Hello world!\n");
  return 0;
}
```

Compiler


High level language
Machine code
Easy for programmer to understand
Contains English words
Translator program
The computer's own language
Binary numbers All 1s and 0s

gcc 컴파일러

- gcc(GNU cc) 컴파일러 상업용 C 컴파일러(cc)
  $ gcc [-옵션] 파일 $ cc [-옵션] 파일

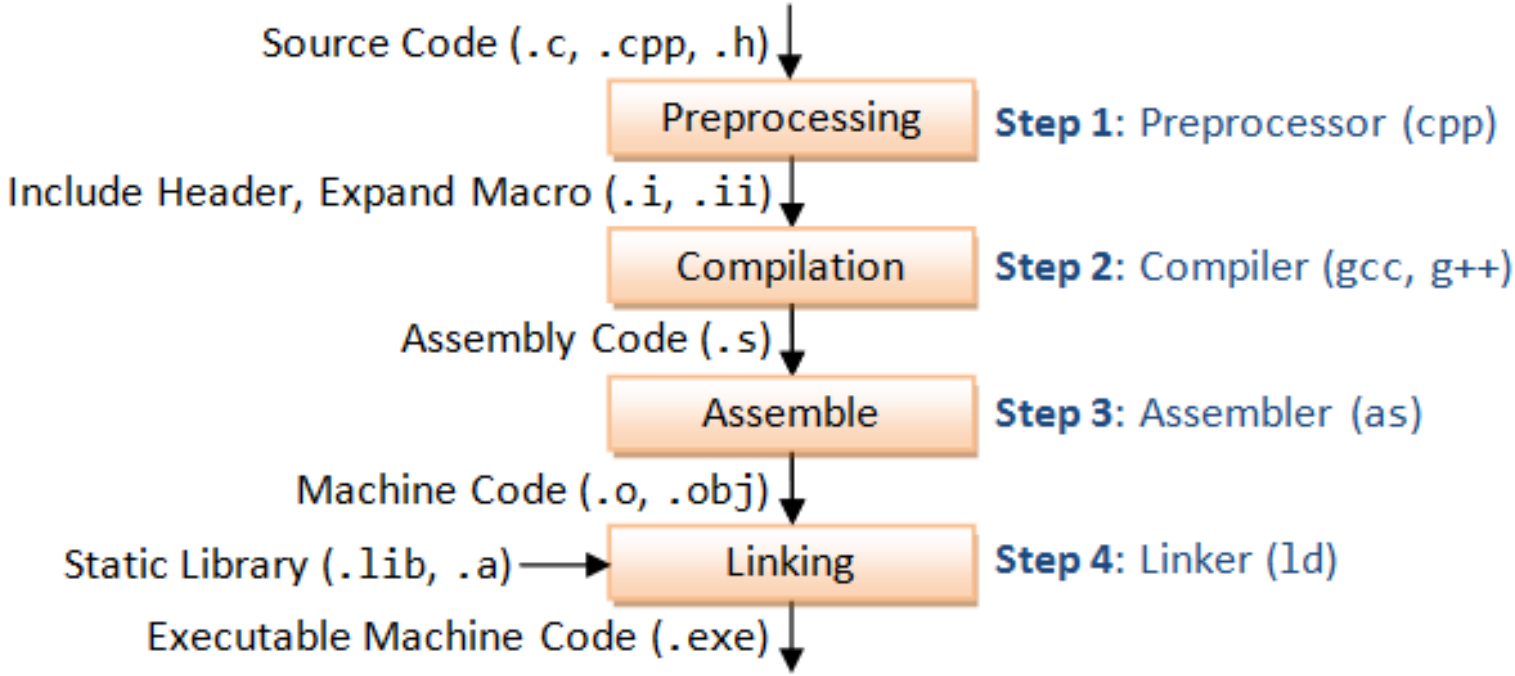
- 컴파일
```
$ gcc longest.c
$ a.out              // 실행
```

- -c 옵션
```
$ gcc -c longest.c
```

- -o 옵션
```
$ gcc -o longest longest.o
혹은
$ gcc -o longest longest.c

$ longest            // 실행
```

Compile Process



Code Review

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
  printf("Hello world!\n");
  return 0;
}
```

- Lines that begin with a # communicate with the preprocessor.
- This #include line causes the preprocessor to include a copy of the header file stdio.h at this point in the code.

Code Review (Cont'd)

```
#include <stdio.h>

int main(int argc, char *argv[])
{
  printf("Hello world!\n");
  return 0;
}
```

- A function prototype has the following general form:

type function_name(parameter type list);

Code Review (Cont'd)

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
  printf("Hello world!\n");
  return 0;
}
```

- The C system provides a standard library of functions:

printf(), scanf(), …

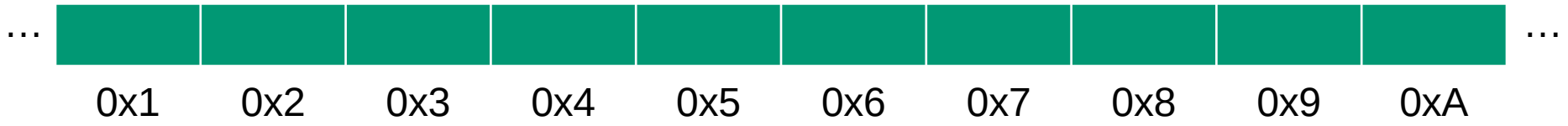
Code Review (Cont'd)

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
  printf("Hello world!\n");
  return 0;
}
```

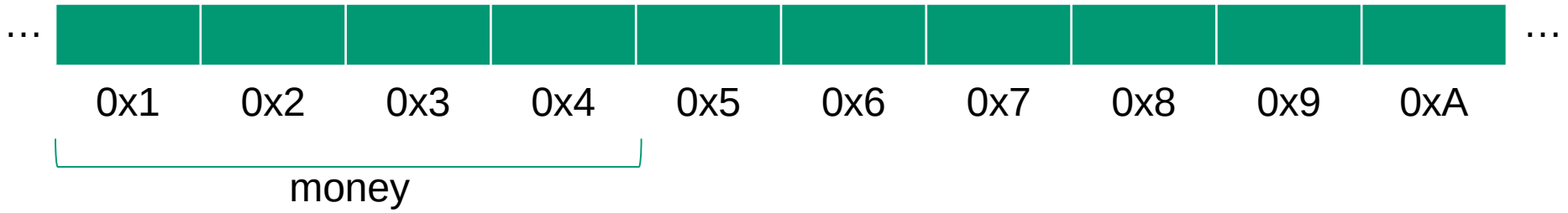
- When a return statement is encountered in a function, control is passed back to the calling environment.

Variables



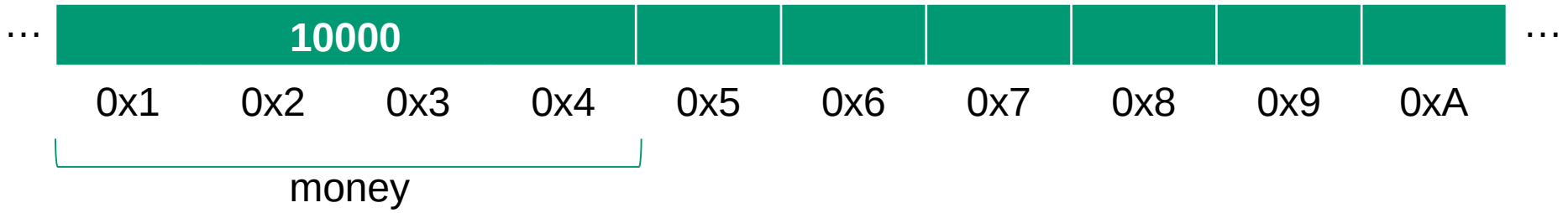

```
int money;
money = 10000;
money = money + 10000;
```

Variables



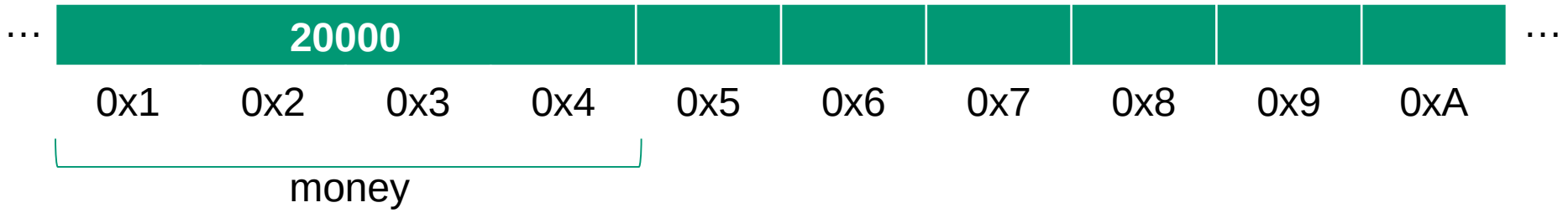

```
int money;
money = 10000;
money = money + 10000;
```

Variables




```
int money;
money = 10000;
money = money + 10000;
```

Variables

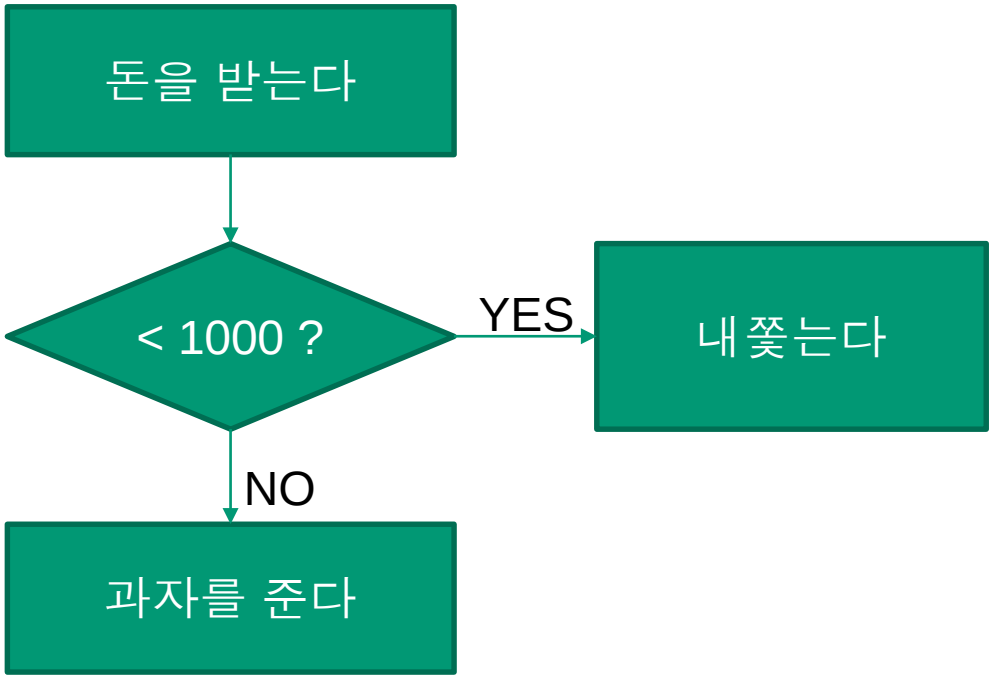



```
int money;
money = 10000;
money = money + 10000;
```

Variables

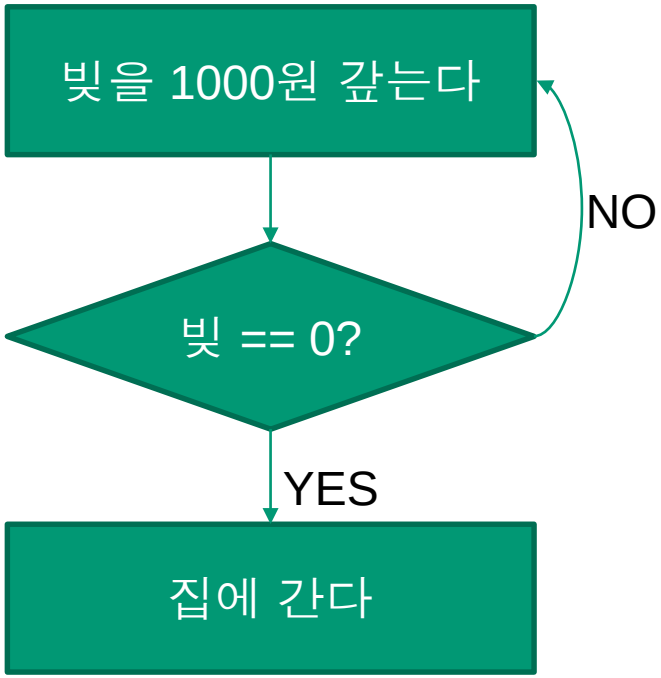
Name	Size*	Range*
char	1byte	signed: -128 to 127 unsigned: 0 to 255
short int (short	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
long int (long)	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
bool	1byte	true or false
float	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	8bytes	+/- 1.7e +/- 308 (~15 digits)

Flow Control - Conditional Statement




```
int money;
scanf("%d", &money);
if(money < 1000)
{
        printf("내쫓는다");
}
else
{
        printf("과자를 준다");
}
```

Flow Control – Loop Statement




```
int loan;

for (loan = 10000; loan != 0; loan=loan-1000)
{
        printf("빚을 1000원 갚는다");
}

printf("집에간다");
```

Functions

```
void main()
{
int num;

num = square(4);

printf("%d",num)
}
```

```
int square(int n1)
{

int x = n1 * n1;

return(x);
}
```