Fundamentals of Programming

Lecture 8

Chamila Karunatilake

Department of Information and Communication Technology

Faculty of Technology

University of Sri Jayewardenepura

chamilakarunatilake@sjp.ac.lk

Types of Variables

There are three types of variables in C programming language

- Variables declared **Inside a function** or a **block** which are called **local** variables.
- Variables declared **outside of all functions** which are called **global** variables.
- Variables declared **in the definition of function** parameters which are called **formal parameters**.

Types of Variables

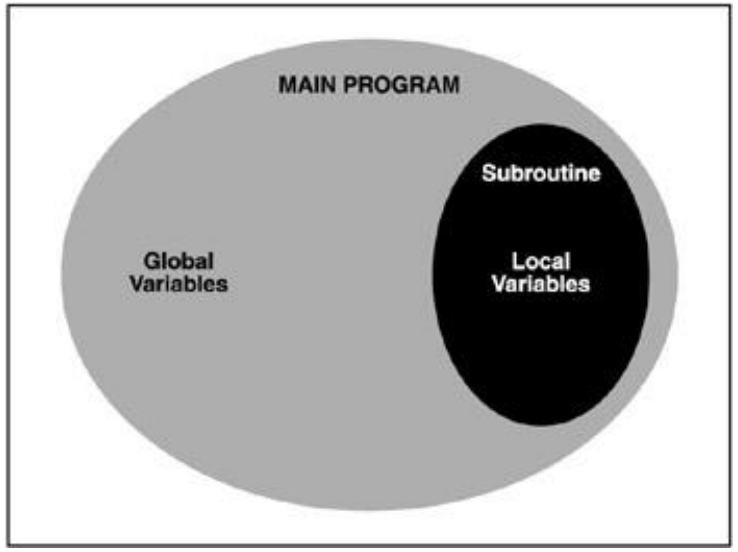
```
#include <stdio.h>

int g; // Global variable

int main () {
   int a, b;
   a = 10;  // local variable
   b = 20;  // local variable
   g = a + b;
   printf("value of a = %d, b = %d and g = %d\n", a, b, g);
   return 0;
}
```

Variable Scope

- A **variable scope** is a **region** or **area** of the program where a defined variable is visible(or has power).
- Beyond that **region**, variable is not visible and from outside, it **cannot be accessed**.



Scope Rules

- **Global** variables are visible to **all the parts of the program**.
- **Local** variables are **only** visible **to the block** of the code where the variable is defined. They cannot be accessed from outside to that block or from other blocks.
- If a local variable is defined inside a function, it is only visible to the function.
- If a local variable is defined inside a for loop, it is only visible to the for loop block.
- **Function parameters** are only visible **to the function**. They cannot be accessed outside the function.

Scope Rules

```
#include <stdio.h>
int main ()
{
   int a = 10;
   int b = 12;
   int sum = add(a,b);
   int sub = subtract(a,b);
   printf("%d/n",sum);
   printf("%d/n",sub);
}
```

```
int add(int x,int y)
{
   int sum = x + y;
   return sum;
}

int subtract(int x,int y)
{
   int sub = x - y;
   return sub;
}
```

Scope Rules

```
#include <stdio.h>
int result = 0;
int main ()
{
   int a = 10;
   int b = 12;
   add(a,b);
   printf("%d/n",result);
   subtract(int x,int y);
   printf("%d/n",result);
}
```

```
void add(int x,int y)
{
   result = x + y;
}

void subtract(int x,int y)
{
   result = x - y;
}
```

Scope Rules

- Global variables should be defined at the top of the program before the first function(which is usually the main() function).
- Otherwise, it will not be visible to the functions above the defined position.

```
#include <stdio.h>
int main ()
{
 ……
}
int g;
int myfunction()
{
   ……
}
```

```
#include <stdio.h>
int g;
int main ()
{
   ……
}
int myfunction()
{
   ……
}
```

Scope Rules

- Local variables are visible only to the block where it is defined.
- Even inside the block, it is not visible to the area above the position where it is defined.

```
void test()
{
    printf("%d\n",x);
    int x = 20;
    printf("%d\n",x * x);
}
```

```
void test()
{
    int x = 20;
    printf("%d\n",x);
    printf("%d\n",x * x);
}
```

Scope Rules

- Multiple variables with the same name cannot be declared in the same scope.
- Multiple variables with the same name can be declared in different scopes.
- If the scopes are independent from each other, those variables are exist separately, independent from each other.
- If scopes are conflicting, the most local variable is visible.
- For example, if there is a global variable on top of the program and a local variable inside a function with the same name.
- Inside the function, local variable is visible, and it covers the visibility of global variable.

Scope Rules

```
#include <stdio.h>
int result = 0;
int main ()
{
   int a = 10;
   int b = 12;
   add(a,b);
   printf("%d/n",result);
   subtract(a,b);
   printf("%d/n",result);
}
```

```
void add(int x,int y)
{
   int result = x + y;
}

void subtract(int x,int y)
{
   int result = x - y;
}
```

Questions?