Short-term Hands-on Supplementary Course on C Programming

Session 2: Conditional Statements

Authors:
Nivedhitha D
Karthik D

May 18, 2022

1 Flow of Control

Every procedural language provides statements for determining the flow of control within programs. The normal flow of control among statements is sequential, proceeding from one statement to the next. However, as we shall see, most of the statements in C are designed to alter this sequential flow so that algorithms of arbitrary complexity can be implemented. This is done with statements that control whether or not other statements execute and, if so, how many times. There are three basic flow control constructs:

1. Sequential
2. Conditional (decision)
3. Loop (iteration)

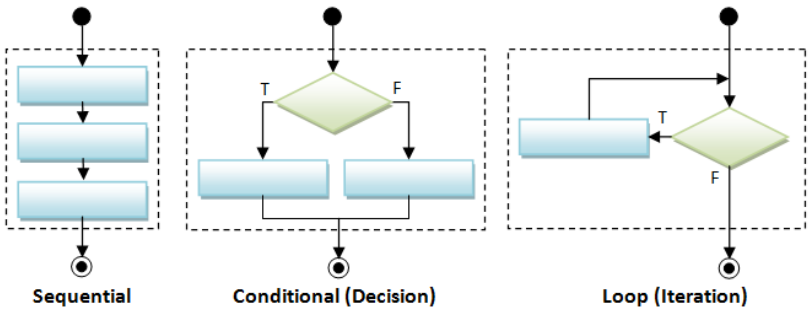

Figure 1: The three basic flow control constructs - sequential, conditional, and loop.

1.1 Sequential Flow Control

A program is a sequence of instructions. Sequential flow is the most common and straight-forward, where programming statements are executed in the order that they are written - from top to bottom in a sequential manner.

```c
/*
* euclidean_distance.c
* This program calculates the euclidean distance
* between two points given by the user.
*/
#include<stdio.h>
#include<math.h>
int main()
{
   int x1, y1, x2, y2, x, y, distance;

   // take first point's coordinates
   printf("Enter coordinates of first point: ");
   scanf("%d %d",&x1, &y1);

   // take second point's coordinates
   printf("Enter coordinates of second point: ");
   scanf("%d %d",&x2, &y2);

   x = (x2-x1);
   y = (y2-y1);

   distance = sqrt(x*x + y*y);

   // display result
   printf("Distance = %d", distance);

   return 0;
}
```

Listing 1: Sequential program execution explained using a program to calculate the euclidean distance between two points in the X-Y plane.

1.2 Conditional (Decision) Flow Control

Decision making structures require that the programmer specifies one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false. C programming language assumes any **non-zero** and **non-null** values as **true**, and if it is either **zero** or **null**, then it is assumed as **false** value.

- if-then
- if-then-else
- if-elseif-elseif-...-else
- switch-case
- Conditional or ternary operator

1.2.1 if-then

```
// if-then
if ( booleanExpression ) {
   true-block ;
}
```

```
if (mark >= 50) {
   printf("Congratulation!\n");
   printf("Keep it up!\n");
}
```

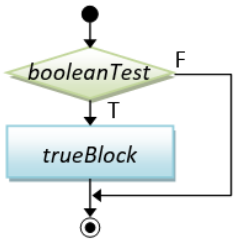

Figure 2: Syntax, example and flowchart of if-then conditional construct.

```c
/*
This program calculates the maximum of three numbers.
*/
#include <stdio.h>

int main() {

    int a, b, c;
    int max;

    printf("Enter a: ");
    scanf("%d", &a);

    printf("Enter b: ");
    scanf("%d", &b);

    printf("Enter c: ");
    scanf("%d", &c);

    max = a;

    if(a > b) {
        if(a > c) {
            max = a;
        }
    }

    if(b > a) {
        if(b > c) {
            max = b;
        }
    }

    if(c > a) {
        if(c > b) {
            max = c;
        }
    }

    printf("The maximum of %d, %d, %d is %d.\n", a, b, c, max);

  return 0;
}
```

Listing 2: if-then nested program execution for max of 3 numbers.

```c
/*
This program calculates the maximum of three numbers.
*/

#include <stdio.h>

int main() {

    int a, b, c;

    int max = a;

    printf("Enter three different numbers: ");
    scanf("%d %d %d", &a, &b, &c);

    // if a is greater than both b and c, a is the largest
    if (a >= b && a >= c)
        max = a;

    // if b is greater than both a and c, b is the largest
    if (b >= a && b >= c)
        max = b;

    // if n3 is greater than both a and b, c is the largest
    if (c >= a && c >= b)
        max = c;

    printf("The maximum of %d, %d, %d is %d.\n", a, b, c, max);

  return 0;
}
```

Listing 3: if-then program execution for max of 3 numbers.

1.2.2 if-then-else

```c
// if-then-else
if ( booleanExpression ) {
   true-block ;
} else {
   false-block ;
}
```

```c
if (mark >= 50) {
   printf("Congratulation!\n");
   printf("Keep it up!\n");
} else {
   printf("Try Harder!\n");
}
```

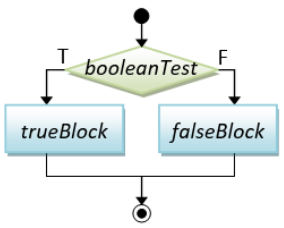

Figure 3: Syntax, example and flowchart of if-then-else conditional construct.

```c
/*
This program calculates the maximum of three numbers.
*/
#include <stdio.h>

int main(void) {

    int a, b, c;
    int max;

    printf("Enter a: ");
    scanf("%d", &a);

    printf("Enter b: ");
    scanf("%d", &b);

    printf("Enter c: ");
    scanf("%d", &c);

    max = a;

    if(a > b && a > c) {
            max = a;
        }

    else {
            if (b > a && b > c) {
                max = b;
            }

            else
                max = c;
        }

    printf("The maximum of %d, %d, %d is %d.\n", a, b, c, max);

  return 0;
}
```

Listing 4: if-then-else program execution for max of 3 numbers.

1.2.3 if-elseif-elseif-...-else

```
// nested-if
if ( booleanExpr-1 ) {
   block-1 ;
} else if ( booleanExpr-2 ) {
   block-2 ;
} else if ( booleanExpr-3 ) {
   block-3 ;
} else if ( booleanExpr-4 ) {
   ......
} else {
   elseBlock ;
}
```

```
if (mark >= 80) {
   printf("A\n");
} else if (mark >= 70) {
   printf("B\n");
} else if (mark >= 60) {
   printf("C\n");
} else if (mark >= 50) {
   printf("D\n");
} else {
   printf("F\n");
}
```

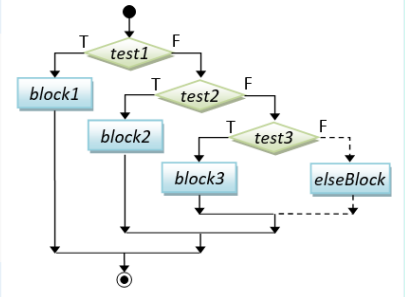

Figure 4: Syntax, example and flowchart of if-elseif-elseif-...-else conditional construct.

```c
/*
This program calculates the maximum of three numbers.
*/
#include <stdio.h>

int main(void) {

    int a, b, c;
    int max;

    printf("Enter a: ");
    scanf("%d", &a);

    printf("Enter b: ");
    scanf("%d", &b);

    printf("Enter c: ");
    scanf("%d", &c);


    if (a > b && a > c) {
        max = a;
    }

    else if (b > a && b > c) {
        max = b;
    }

    else if (c > a && c > b) {
        max = c;
    }

    // a = b = c
    else {
     max = a;
    }


    printf("The maximum of %d, %d, %d is %d.\n", a, b, c, max);

  return 0;
}
```

Listing 5: if-elseif-else program execution for max of 3 numbers.

1.2.4 switch-case

```
// switch-case
switch ( selector ) {
   case value-1:
      block-1; break;
   case value-2:
      block-2; break;
   case value-3:
      block-3; break;
   ......
   case value-n:
      block-n; break;
   default:
      default-block;
}
```

```
char oper; int num1, num2, result;
......
switch (oper) {
   case '+':
      result = num1 + num2; break;
   case '-':
      result = num1 - num2; break;
   case '*':
      result = num1 * num2; break;
   case '/':
      result = num1 / num2; break;
   default:
      printf("Unknown operator\n");
}
```

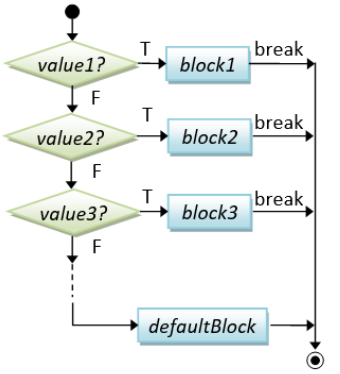

Figure 5: Syntax, example and flowchart of switch-case conditional construct.

```c
#include <stdio.h>
int main()
{
    int week;
    printf("Enter week number(1-7): ");
    scanf("%d", &week);

    switch(week)
    {
        case 1:
            printf("Sunday");
            break;
        case 2:
            printf("Monday");
            break;
        case 3:
            printf("Tuesday");
            break;
        case 4:
            printf("Wednesday");
            break;
        case 5:
            printf("Thursday");
            break;
        case 6:
            printf("Friday");
            break;
        case 7:
            printf("Saturday");
            break;
        default:
            printf("Invalid input! Please enter week number between 1-7.");
    }
    return 0;
}
```

Listing 6: switch case program to print day of week.

1.2.5 Conditional or Ternary Operator

```
booleanExpr ? trueExpr : falseExpr
```

```
printf("%s\n", (mark >= 50) ? "PASS" : "FAIL");
   // print either "PASS" or "FAIL"
max = (a > b) ? a : b;   // RHS returns a or b
abs = (a > 0) ? a : -a;  // RHS returns a or -a
```

Figure 6: Syntax and example for the conditional or ternary operator.

```
/*
This program calculates the maximum of three numbers.
*/
#include <stdio.h>

int main(void) {

    int a, b, c;
    int max;

    printf("Enter a: ");
    scanf("%d", &a);

    printf("Enter b: ");
    scanf("%d", &b);

    printf("Enter c: ");
    scanf("%d", &c);

    max = (a > b) ? a : b;
    max = (max > c) ? max : c;

    // max = (a > b && a > c) ? a: (b > a && b > c) ? b : (c > a &&
     c > b) ? c : a;

    printf("The maximum of %d, %d, %d is %d.\n", a, b, c, max);

  return 0;
}
```

Listing 7: Ternary operator program execution for max of 3 numbers.

1.3 Loop (Iteration) Flow Control

To be covered in the next session.

2 TUTORIAL: Simple Calculator

Write the code to build a simple calculator in C. Answer the following questions sequentially to build your application.

- What is the input and formatting?
- What is the output and formatting?
- Which of the known conditional constructs is the best? Why?
- Are there any edge cases that may break the code?

```c
/**
 * C program to create Simple Calculator using switch case
 */

#include <stdio.h>

int main()
{
    char op;
    float num1, num2, result=0.0f;

    /* Print welcome message */
    printf("WELCOME TO SIMPLE CALCULATOR\n");
    printf("-----------------------------------\n");
    printf("Enter [number 1] [+ - * /] [number 2]\n");

    /* Input two number and operator from user */
    scanf("%f %c %f", &num1, &op, &num2);

    /* Switch the value and perform action based on operator*/
    switch(op)
    {
        case '+':
            result = num1 + num2;
            break;

        case '-':
            result = num1 - num2;
            break;

        case '*':
            result = num1 * num2;
            break;

        case '/':
            result = num1 / num2;
            break;

        default:
            printf("Invalid operator");
    }

    /* Prints the result */
    printf("%.2f %c %.2f = %.2f", num1, op, num2, result);

    return 0;
}
```

Listing 8: Simple calculator app.

3 PROBLEMS

3.1 Problem 1

Write a program to calculate the shake probability of a Pokeball. Here, a is the catch rate and needs to be read in as the user input. a can be any integral value in the range [1-255]. This equation is given by:

$$b = \lfloor \frac{65536}{\sqrt{\sqrt{\frac{255}{a}}}} \rfloor$$

3.2 Problem 2

Evaluate the following expressions in order and determine the final output.

```

int a=5;
int b=a++;

printf("%d %d %d", --a, b--, ++b);
```

3.3 Problem 3

What will be the output of the following code snippet?

```
#include<stdio.h>
int main()
{
  int a = 555;
  a = -10, -11, -12;
  printf("%d", a);
  return 0;
}
```

3.4 Problem 4a

Write a program to swap the values of two variables using a third variable.

3.5 Problem 4b

Write a program to swap the values of two variables without using a third variable.

3.6 Problem 5

Predict the output of the following code snippet.

```
#include <stdio.h>

int main() {
    // Write C code here
    int a = 2;
    int val;
    val = a == 0 ? 1:
          a == 1 ? 2:
```

```
            4;
    printf("%d", val);
    return 0;
}
```

3.7 Problem 6

Predict the output of the following code snippet.

```
#include <stdio.h>
int main(void)
{
  //variables
  int x, a, b;

  //initialise
  a = 4;
  b = 2;

  //compute x
  x = a * a - 3 * b + a / b;

  //output
  printf("x = %d\n", x);

  printf("End of code\n");
  return 0;
}
```