To:
Tenouk

C LAB WORKSHEET 8b
C & C++ Selection: The Conditional Operator And switch-case-break Part 5

Items in this page:

1. The ?:, conditional operator.
2. The switch-case-break construct and flowcharts.
3. Activities, questions and answers.
4. Tutorial reference that should be used together with this worksheet are: C & C++ program control 1 and C & C++ program control 2.

More Questions and Activities

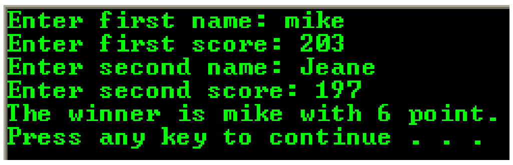
1. Write a complete working program that will ask for a person's name and his/her game score. Then it will ask for a second person's name and score. The program will print the winner's name and also print by how many points that person won.

   The following is a sample of a pseudocode. It is edited many times to achieve the desired program output.

```
Declare int variables to store the scores
    int k, k1;
Declare string variables to store the names
    char name[20] = "Dummy", name1[20] =
"Dummy";
Read & store the first name and score
    printf("Enter first name: ");
    scanf_s("%s", name, 19);
    printf("Enter first score: ");
    scanf_s("%d", &k, 1);
Read & store the second name and score
    printf("Enter second name: ");
    scanf_s("%s", name1, 19);
    printf("Enter second score: ");
    scanf_s("%d", &k1, 1);
Do the comparison, which one is larger
    if(k > k1)
Print the winner name and the point
    else
Print the winner name and the point
Program stop.
```

```
#include <stdio.h>

int main(void)
{
    // variables to store the scores
    int k, k1;
    // strings to store names
    char name[20] = "Dummy", name1[20] =
"Dummy";
    // read & store the first name and score
    printf("Enter first name: ");
    scanf_s("%s", name, 19);
    printf("Enter first score: ");
    scanf_s("%d", &k, 1);
    // read & store the second name and score
    printf("Enter second name: ");
    scanf_s("%s", name1, 19);
    printf("Enter second score: ");
    scanf_s("%d", &k1, 1);
    // if k > k1
    if(k > k1)
    // then
        printf("The winner is %s with %d point.
\n", name, k-k1);
    // else
    else
        printf("The winner is %s with %d point.
\n", name1, k1-k);
    return 0;
}
```



```
Enter first name: mike
Enter first score: 203
Enter second name: Jeane
Enter second score: 197
The winner is mike with 6 point.
Press any key to continue . . .
```

2. Write a program that will read a float and a character for each scanf_s(). The character could be d for deposit or w for withdrawal. Starting with a balance of zero, add the deposits and subtract the withdrawals until the balance becomes negative. Then print by how much the balance went negative. The sample input and output are given below.

```
-----------------Output---------------
100.0    d
20.0     d
60.0     w
200.0    w
Your account now is -140.00 dollars.
```

The following is a pseudocode example for this question.

Declare variables for the amount in float, balance in float and transaction code in char types.

```
    float amount, balance=0.0;
    char transaction;
```

Read & store the amount and transaction code.

```
    scanf_s("%f %c", &amount, &transaction, sizeof
(float), sizeof(char));
```

Test the transaction code, if ‘d’ it is deposit.

```
    if(transaction == 'd')
```

Test the balance so that it is not negative.

```
    for(;balance >=0;)
```

If the balance is not negative or positive sum-up the balance.

```
    balance = balance + amount;
```

And keep reading the next input of amount and code.

```
    scanf_s("%f %c", &amount, &transaction, sizeof
(float), sizeof(char));
```

At the same time keep reading the transaction code for withdrawal, ‘w’. If the transaction code is ‘w’

```
    if(transaction == 'w')
```

Minus the withdrawn amount to update the balance.

```
    balance = balance - amount;
```

Keep reading and storing the withdrawn amount using the do-while because we need to terminate the loop without printing the last input. That is checking the balance for negative at the end of the loop.

```
    do
    {
        scanf_s("%f %c", &amount, &transaction,
sizeof(float), sizeof(char));
        balance = balance - amount;
    }while(balance >=0);
```

When the balance is negative, stop the loop and print the negative balance.

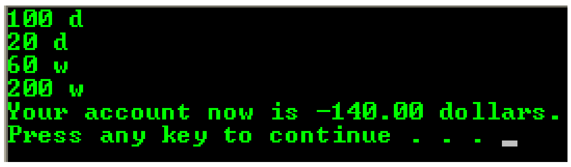
```
    printf("Your account now is %.2f dollars.\n",
balance);
```

Program stop.

```c
#include <stdio.h>

int main(void)
{
    float amount, balance=0.0;
    char transaction;

    scanf_s("%f %c", &amount, &transaction,
sizeof(float), sizeof(char));
    if(transaction == 'd')
    for(;balance >=0;)
    {
        balance = balance + amount;
        scanf_s("%f %c", &amount, &transaction,
sizeof(float), sizeof(char));
        if(transaction == 'w')
        {
            balance = balance - amount;
            do
            {
                scanf_s("%f %c", &amount,
&transaction, sizeof(float), sizeof(char));
                balance = balance - amount;
            }while(balance >=0);
        }
      }
    printf("Your account now is %.2f dollars.\n",
balance);
    return 0;
}
```



```
100 d
20 d
60 w
200 w
Your account now is -140.00 dollars.
Press any key to continue . . .
```

3. Men and women are running races. We need to know what was the lowest men's score and the highest women's score. Data are given so that men's scores alternate with those of women: first men's, then women's. The program stops when a time of zero is read for a man's score. A sample input and output are given below.

```
-----------------Output----------------
7.80
5.70
5.80
8.95
0.0
Lowest men's: 5.80
Highest women's: 8.95
```

The following is a pseudocode example.

Declare variables to store men's and women's scores, lowest men's score and highest women's score and for the array's index.

```
   int i = 0;
   float man_score[20], women_score[20],
lowest_men, highest_women;
```

Begin reading & storing men's score in the man_score array.

```
   scanf_s("%f", &man_score[i], sizeof
(man_score));
```

Assign the first element to the lowest_man variable to do the comparison for the lowest score.

```
   lowest_man = man_score[i];
```

Begin reading & storing women's score.

```
   scanf_s("%f", &women_score[i], sizeof
(women_score));
```

Assign the first element to the highest_women variable to do the comparison for the highest score.

```
   highest_women = women_score[i];
```

Keep reading and storing the next man's and women's scores while man_score[i] != 0.0.

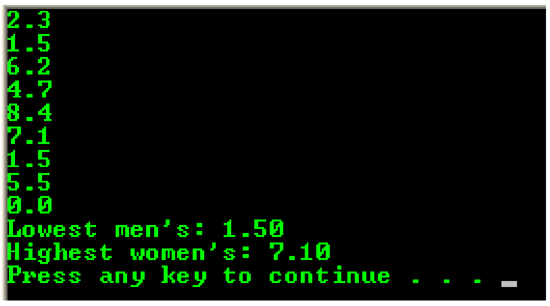
```
for(;man_score[i] != 0.0;)
{
   scanf_s("%f", &man_score[i], sizeof
(man_score));
   scanf_s("%f", &women_score[i], sizeof
(women_score));
}
```

For every array's element, find the lowest man and the highest women scores in the for loop. Assign the highest women score to highest_women variable and lowest man score to lowest_man variable. Don't forget to increment the index for every iteration.

```
for(;man_score[i] != 0.0;)
{
   if(man_score[i] < lowest_man)
      lowest_man = man_score[i];
   if(women_score[i] > highest_women)
      highest_women = women_score[i];
   i++;
   scanf_s("%f", &man_score[i], sizeof
```

```c
#include <stdio.h>

int main(void)
{
   int i=0;
   float man_score[20], women_score[20],
lowest_man, highest_women;
   scanf_s("%f", &man_score[i], sizeof
(man_score));
   lowest_man = man_score[i];
   scanf_s("%f", &women_score[i], sizeof
(women_score));
   highest_women = women_score[i];
   for(;man_score[i] != 0.0;)
   {
     if(man_score[i] < lowest_man)
       lowest_man = man_score[i];
     if(women_score[i] > highest_women)
       highest_women = women_score[i];
     i++;
     scanf_s("%f", &man_score[i], sizeof
(man_score));
     if(man_score[i] == 0.0)
break;
scanf_s("%f", &women_score[i], sizeof
(women_score));
}
printf("Lowest men's: %.2f\nHighest women's:
%.2f\n", lowest_man, highest_women);
return 0;
}
```



```
2.3
1.5
6.2
4.7
8.4
7.1
1.5
5.5
0.0
Lowest men's: 1.50
Highest women's: 7.10
Press any key to continue . . .
```

```
(man_score));
   scanf_s("%f", &women_score[i], sizeof
(women_score));
}
```

The loop need to be terminated when the man_score == 0.0 without reading the next women score. So we can use a break statement here.

```
for(;man_score[i] != 0.0;)
{
   if(man_score[i] < lowest_man)
       lowest_man = man_score[i];
   if(women_score[i] > highest_women)
       highest_women = women_score[i];
   i++;
   scanf_s("%f", &man_score[i], sizeof
(man_score));
   if(man_score[i] == 0.0)
      break;
   scanf_s("%f", &women_score[i], sizeof
(women_score));
}
```

Print the result of the highest women's score and the lowest man's score.

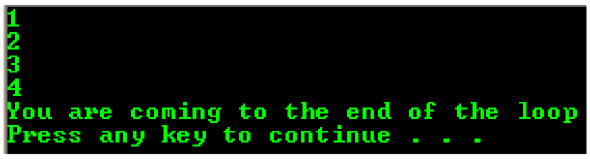
```
   printf("Lowest men's: %.2f\nHighest women's:
%.2f\n", lowest_man, highest_women);
```

Program stop.

4. Ty running the following program.

```
#include <stdio.h>

int main(void)
{
  int i;
  for(i = 1; 5 - i; i = i + 1)
       printf("%d\n", i);
  printf("You are coming to the end of the loop\n");
  return 0;
}
```



```
1
2
3
4
You are coming to the end of the loop
Press any key to continue . . .
```

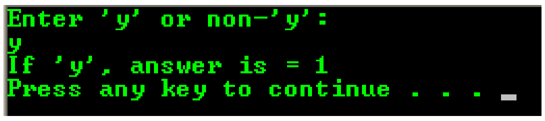
The loop runs only 4 times and then stops. What is really happened here in the loop is that 5 – i is not a condition at all, it is **arithmetic expression**. How is that interpreted? Whenever an arithmetic expression **evaluates to zero**, that expression is considered to have a **false** value. Conversely, whenever an expression is **non-zero**, it is considered to be **true**. Hence, for the first four values of i, 5 – i has a non-zero or a true value. When i becomes 5, then 5 – i becomes zero, the expression becomes false and the loop stops.
Writing condition like this takes fewer instruction cycles from CPU/processor and the code runs faster than the if conditional operators were used.

The Conditional Operator (?:), ternary

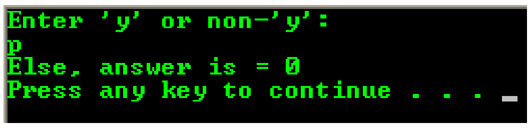
- Whenever one value has to be assigned to a variable if a certain condition is true and another has to be assigned to that same variable if it is false, then the conditional operator can be used. Using the conditional operator will simplify your coding. The following if-else example is converted to using ?: operator.

```
#include <stdio.h>

int main(void)
{
// initialize with dummy value...
char x = 'n';
printf("Enter 'y' or non-'y': \n");
scanf_s(" %c", &x);
// the following codes can be converted using ?: operator
if(x == 'y')
            printf("If 'y', answer is = 1\n");
        else
         printf("Else, answer is = 0\n");
      return 0;
}
```



```
Enter 'y' or non-'y':
y
If 'y', answer is = 1
Press any key to continue . . .
```

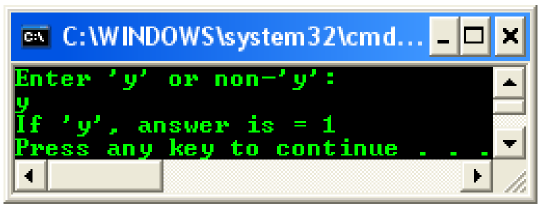


```
Enter 'y' or non-'y':
p
Else, answer is = 0
Press any key to continue . . .
```

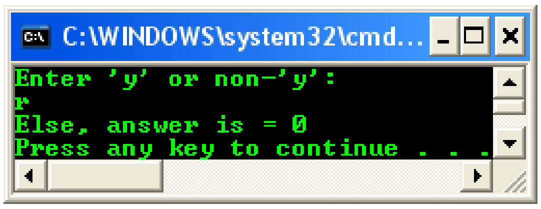
A sample input and output are given below.

```
#include <stdio.h>

int main(void)
{
    // initialize with dummy value...
    char x = 'n';
    printf("Enter 'y' or non-'y': \n");
    scanf_s(" %c", &x);
    // using ?: operator
    (x == 'y') ? printf("If 'y', answer is = 1\n") : printf("Else, answer is
= 0\n");
    return 0;
}
```



```
Enter 'y' or non-'y':
y
If 'y', answer is = 1
Press any key to continue . . .
```



```
Enter 'y' or non-'y':
r
Else, answer is = 0
Press any key to continue . . .
```

From the program example, if x == 'y' is **true**, the following printf() will be executed.

```
printf("If 'y', answer is = 1\n")
```

Otherwise the second printf() will be executed.

```
printf("Else, answer is = 0\n")
```

The switch-case-break Statement

- An organized way to write if statements that depend on the value of one variable is to use the switch-case-break statement. The following example shows how the if program is converted to using the switch statement. The breaks are required for all the cases except for the last one. Also the default case is optional.

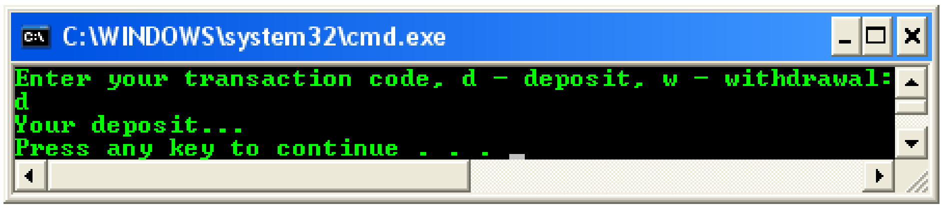
```c
#include <stdio.h>

int main(void)
{
      char code = 'd';
      float balance = 0.0, amount = 0.0;
      printf("Enter your transaction code, d - deposit, w - withdrawal:
\n");
      scanf_s(" %c", &code);
      // the following code can be converted to using
      // switch-case-break statement...
      if(code == 'd')
      {
            printf("Your deposit...\n");
            balance = balance + amount;
      }
      else if(code == 'w')
      {
           printf("Your withdrawal...\n");
           balance = balance - amount;
      }
      else
          printf(" %c code not allowed\n", code);
      return 0;
}
```

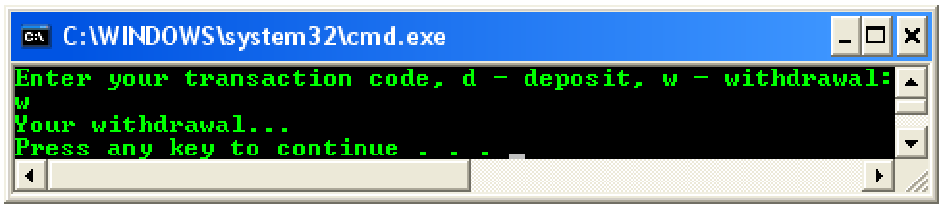
```c
#include <stdio.h>

int main(void)
{
      char code = 'd';
      float balance = 0.0, amount = 0.0;
      printf("Enter your transaction code, d - deposit, w -
withdrawal: \n");
      scanf_s(" %c", &code);
      // switch-case-break statement...
      switch(code)
      {
            case 'd':
                  {
                        printf("Your deposit...\n");
                        balance = balance + amount;
                        break;
                  }
            case 'w':
                  {
                        printf("Your withdrawal...\n");
                        balance = balance - amount;
                        break;
                  }
            default:
                  printf("%c code not allowed. Try again!\n", code);
      }
      return 0;
}
```

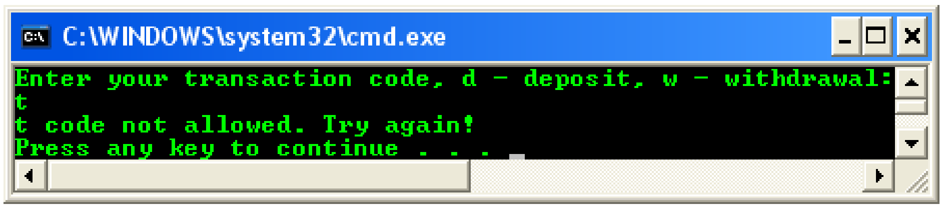
- A sample input and output are given below.



```
C:\WINDOWS\system32\cmd.exe
Enter your transaction code, d - deposit, w - withdrawal:
d
Your deposit...
Press any key to continue . . .
```

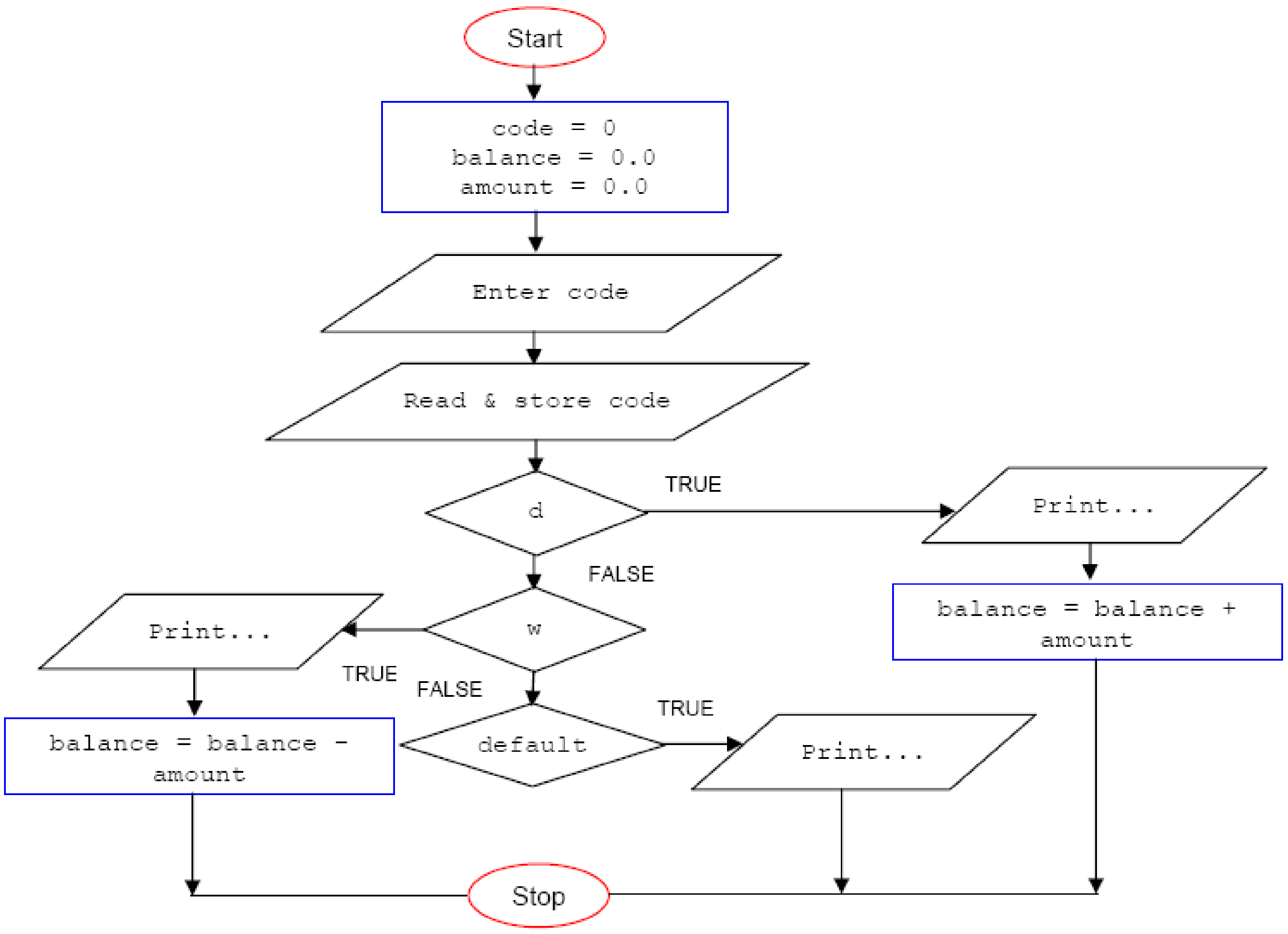


```
C:\WINDOWS\system32\cmd.exe
Enter your transaction code, d - deposit, w - withdrawal:
w
Your withdrawal...
Press any key to continue . . .
```



```
C:\WINDOWS\system32\cmd.exe
Enter your transaction code, d - deposit, w - withdrawal:
t
t code not allowed. Try again!
Press any key to continue . . .
```

- Try building a flowchart for this switch-case-break program example.



More Activities

1. In the following practice, we will use the data in Table 3. We will use the rate in the Table to construct the logic for determining the correct rate. ‘r’ will stand for “residential”, ‘c’ will stand for “commercial” and ‘u’ stand for “unit”.

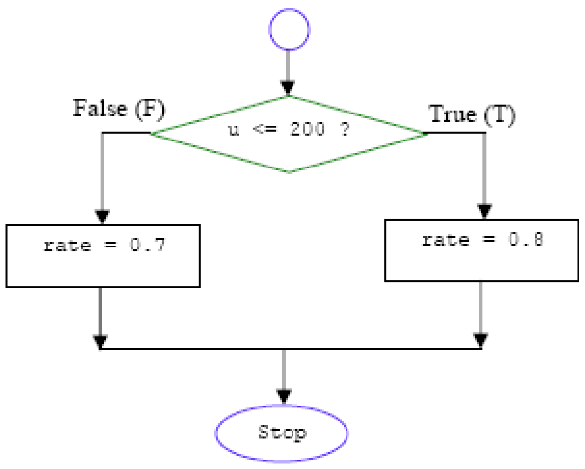
Units used	Residential	Commercial
0 < units <= 200	0.8	0.6
above 200	0.7	0.3

Table 3

Let us first consider the condition for the case where type is equal to “residential” only. If the type is residential and unit of consumption are less than or equal to 200, then the rate is 0.8, according to the Table. If the units are not less than or equal to 200, then the rate is made equal to 0.7. Complete the coding for this part of the logic.

```
if(u <= 200)
    rate = 0.8;
```

```
else
    rate = 0.7;
```



```
if(u <= 200)
    rate = ___________________;
else
    rate = ___________________;
```

| Main |< C & C++ if and if-else 4 | C & C++ switch-case-break 6 >| Site Index | Download |

The C Selection if, if-else, if-else-if, break, conditional/ternary operator and switch-case-break: Part 1 | Part 2 | Part 3 | Part 4 | Part 5 | Part 6