PROGRAMMING

CT103

Week 4a

Note on Lab Assignments

- Please submit your assignments on time.
- Late submissions will be deducted marks.

Lecture Content

- Last lecture (Week 3b):
  - Loops.
  - While loops.
  - Do while loops.
- Today's lecture (Week 4a):
  - Recap on while loops.
  - For loops.
  - Example C program.

WHILE LOOPS RECAP

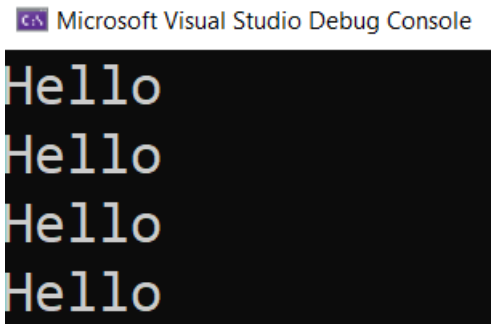
While Loops

- Last week we learned about **while** and **do while** loops.
- The while loop will repeat a block of code over and over while some condition is true.

While Loops

- See the output of this code:

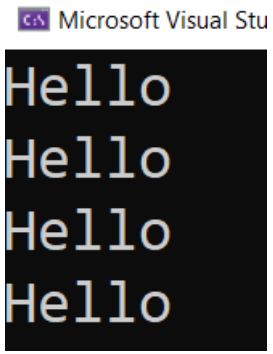
```
int j = 0;
while (j<4) {
    printf("Hello\n");
    j++;
}
```



Do While Loops

- Print “Hello” 4 times using a do while loop.

```
int j = 0;
do {
    printf("Hello\n");
    j++;
} while (j<4);
```



Another Do While Example

- Write a program that:
  - Asks the user if they want to convert a temperature from Celsius to Fahrenheit or the other way around.
  - The program should convert temperature from one unit to the other, e.g. Fahrenheit to Celsius.
  - The user should be able to do as many temperature conversions as they like.

Another Do While Example

```c
#include <stdio.h>
void main()
{
    double temp1;
    double temp2;
    int choice = 1;

    printf("enter choice\n1-Fahrenheit to Celsius\n2-Celsius to Fahrenheit\n3-Exit\n");
    scanf_s("%d", &choice);
    do {
        printf("Enter temp: ");
        scanf_s("%lf", &temp1);

        if (choice == 1)
        {
            temp2 = (temp1 - 32.0) * 5.0 / 9.0;
            printf("%.2lf degrees F = %.2lf degrees C\n\n", temp1, temp2);
        }
        else
        {
            temp2 = 32.0 + (temp1 * 9.0 / 5.0);
            printf("%.2lf degrees C = %.2lf degrees F\n\n", temp1, temp2);
        }

        printf("enter choice\n1-Fahrenheit to Celsius\n2-Celsius to Fahrenheit\n3-Exit\n");
        scanf_s("%d", &choice);

    } while (choice!=3);
}
```

Another Do While Example

```c
#include <stdio.h>
void main()
{
    double temp1;
    double temp2;
    int choice = 1;

    printf("enter choice\n1-Fahrenheit to Celsius\n2-Celsius to Fahrenheit\n3-Exit\n");
    scanf_s("%d", &choice);
    do {
        printf("Enter temp: ");
        scanf_s("%lf", &temp1);

        if (choice == 1)
        {
            temp2 = (temp1 - 32.0) * 5.0 / 9.0;
            printf("%.2lf degrees F = %.2lf degrees C\n\n", temp1, temp2);
        }
        else
        {
            temp2 = 32.0 + (temp1 * 9.0 / 5.0);
            printf("%.2lf degrees C = %.2lf degrees F\n\n", temp1, temp2);
        }

        printf("enter choice\n1-Fahrenheit to Celsius\n2-Celsius to Fahrenheit\n3-Exit\n");
        scanf_s("%d", &choice);

    } while (choice!=3);
}
```

Microsoft Visual Studio Debug Console

```
enter choice
1-Fahrenheit to Celsius
2-Celsius to Fahrenheit
3-Exit
2
Enter temp: 18.3
18.30 degrees C = 64.94 degrees F

enter choice
1-Fahrenheit to Celsius
2-Celsius to Fahrenheit
3-Exit
3
```

Another Do While Example

```c
#include <stdio.h>
void main()
{
    double temp1;
    double temp2;
    int choice = 1;

    printf("enter choice\n1-Fahrenheit to Celsius\n2-Celsius to Fahrenheit\n3-Exit\n");
    scanf_s("%d", &choice);
    do {
        printf("Enter temp: ");
        scanf_s("%lf", &temp1);

        if (choice == 1)
        {
            temp2 = (temp1 - 32.0) * 5.0 / 9.0;
            printf("%.2lf degrees F = %.2lf degrees C\n\n", temp1, temp2);
        }
        else
        {
            temp2 = 32.0 + (temp1 * 9.0 / 5.0);
            printf("%.2lf degrees C = %.2lf degrees F\n\n", temp1, temp2);
        }

        printf("enter choice\n1-Fahrenheit to Celsius\n2-Celsius to Fahrenheit\n3-Exit\n");
        scanf_s("%d", &choice);

    } while (choice!=3);
}
```

There are a few problems with this code.

Can you spot them?

Points to Remember

- Loops allow us to repeat a piece of code.
- While loops allow us to keep repeating as long as the condition is true.
- Avoid infinite loops. Do this by changing "*something*" in the body of the loop.

Points to Remember

- Do while loops are similar to while loops except that they ensure what is in body of loop is executed at least once.

- Loops allow us to have shorter and more readable code.

FOR LOOPS

For Loops

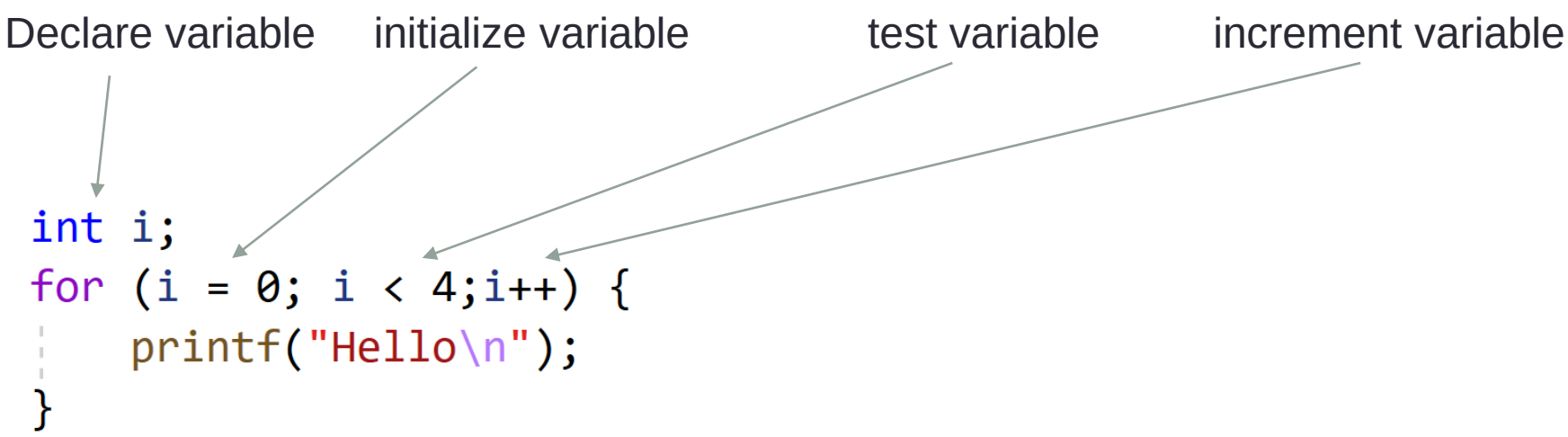
- **For loops** are useful if we want to repeat some code a predetermined number of times.

- We saw how we do this with **while loops**. We use:
  - A variable.
  - A condition.
  - Some change to the variable.

- **For loops** are a another way of doing this!

For Loop Template

- So what does a **for loop** look like?
- A **for loop** will look something like the following:



For Loop Template

- A **for loop** can also look as follows:

```
int i;
for (i = 0; i < 4;i++) {
    printf("Hello\n");
}
```

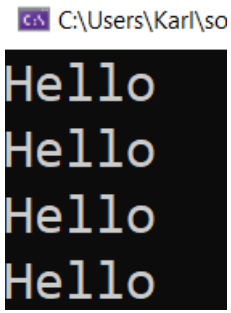
```
for (int i = 0; i < 4;i++) {
    printf("Hello\n");
}
```

- We can declare and initialize the variable in the for loop.

For Loop Example

- When we run the program, it outputs “Hello” 4 times:

```
int i;
for (i = 0; i < 4;i++) {
    printf("Hello\n");
}
```



For Loop vs While Loop

- Lets compare the structure of for loops and while loops.

```c
for (int i = 0; i < 4;i++) {
    printf("Hello\n");
}
```

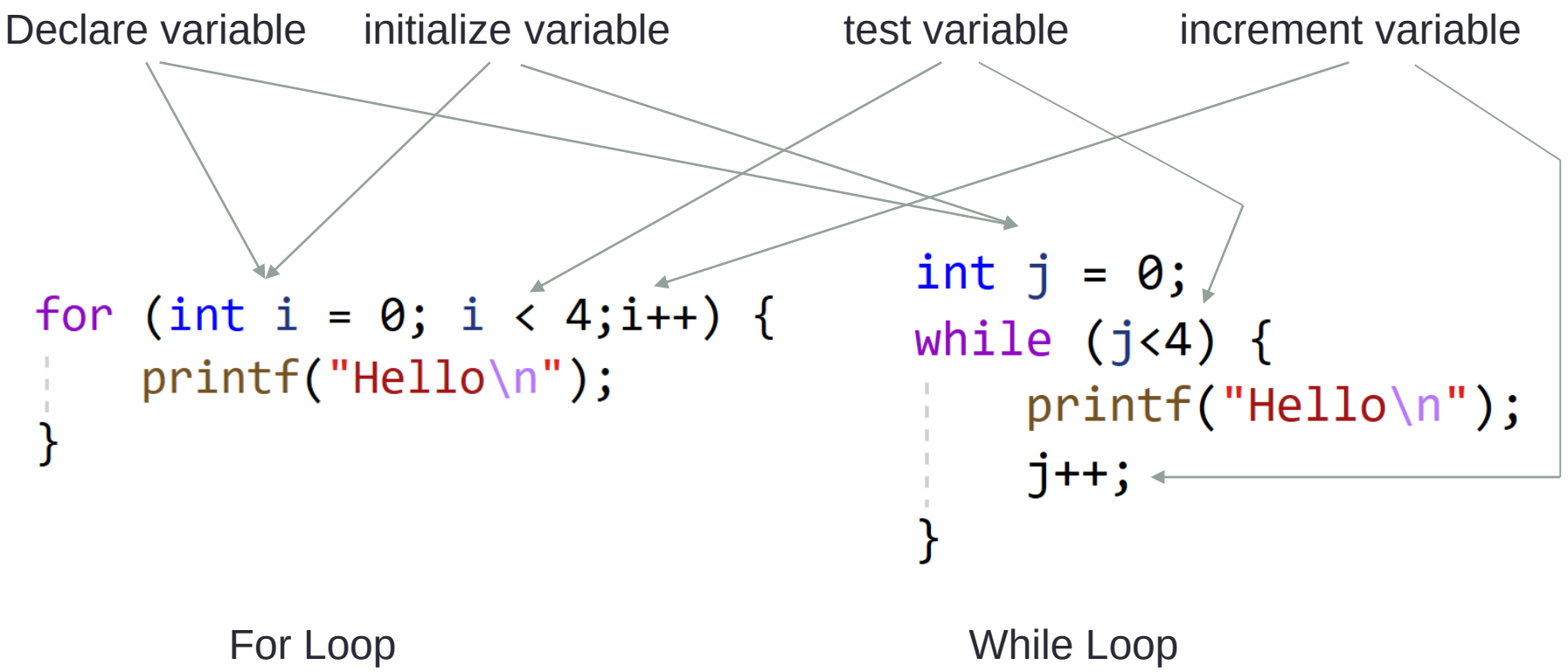
For Loop

```c
int j = 0;
while (j<4) {
    printf("Hello\n");
    j++;
}
```

While Loop

For Loop vs While Loop

- Lets compare the structure of for loops and while loops.



Which Loop Should I Use?

- Your choice of loop depends on what you want to do and how you want to end the loop.

- If you want to repeat a task “x” number of times, you can use either a **for loop** or a **while loop**.
  - E.g. If I want to do a calculation 5 times, use a for loop.
  - When I want to end the loop is determined by the number of calculations.

- If you do not know how many cycles the loop will run for, use a **while loop**.
  - E.g. If my program does BMI calculations, I don’t know how many calculations the user will want to do.

EXAMPLE PROBLEMS

ATM Problem

- You are working for a bank.
- You must write a program that:
  - Create a new bank account with a balance of €100.
  - Use a for loop to make 3 ATM withdrawals.
  - Update the bank account balance for each withdrawal.

ATM Problem

- Go to C program solution.

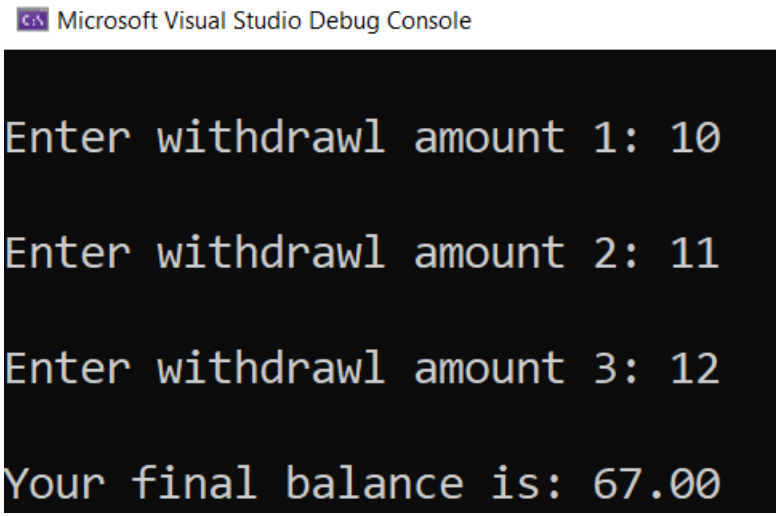
ATM Problem

- The following code will work:

```
#include <stdio.h>
void main()
{
    float balance = 100.0;
    float withdraw = 0.0;
    for (int i = 0; i < 3;i++) {
        printf("\nEnter withdrawl amount %d: ",i+1);
        scanf_s("%f",&withdraw);
        balance = balance - withdraw;
    }
    printf("\nYour final balance is: %0.2f",balance);
}
```

ATM C Program Output

- The code will produce the following output:



Airlines Problem

- You are working for a major airline “Brianair”.
- You must write a program that:
  - Reads in the number of bags to be checked in as input from the user.
  - Use a for loop to read in the weight of each individual bag.
  - Sum up the total weight of the bags and print it to the screen.
  - Rewrite the same program as outlined above, now using a while loop.

Airlines Problem

- Go to C program solution.

Airlines Problem

- The following code will work:

```
#include <stdio.h>
void main()
{
    int numBags = 0;
    int counter;
    float bagWeight;
    float totalWeight = 0;
    printf("Enter the number of bags: ");
    scanf_s("%d",&numBags);
    for (counter = 0; counter < numBags;counter++) {
        printf("Enter the weight of bag %d: ",counter+1);
        scanf_s("%f",&bagWeight);
        totalWeight = totalWeight + bagWeight;
    }
    printf("Total weight = %0.2f ",totalWeight);
}
```

Airlines Problem

- The following code will also work, now with a while loop:

```
#include <stdio.h>
void main()
{
    int numBags = 0;
    int counter = 0;
    float bagWeight;
    float totalWeight = 0;
    printf("Enter the number of bags: ");
    scanf_s("%d",&numBags);
    while (counter < numBags) {
        printf("Enter the weight of bag %d: ",counter+1);
        scanf_s("%f",&bagWeight);
        totalWeight = totalWeight + bagWeight;
        counter++;
    }
    printf("Total weight = %0.2f ",totalWeight);
}
```

Airline C Program Output

- The code will produce the following output: