1. Wiederholung

Schreibe ein **HelloWorld** Programm

- Gebe ", \ und eine **leere Zeile** aus

Lösung:

Lösung 1: Hello World 2.0

```c
#include <stdio.h>
int main(int argc, char **argv)
{
  printf("Hello World!\n");
  printf("Name: \"Mein Name\"\n"); // Hochkomma
  printf("\n"); // Neue Zeile
  printf("C:\\Windows\\\n"); // Backslash
  return 0;
}
```

2. dynamische Ausgabe

Das Kommando **printf** benutzt das %-Zeichen zur Markierung welcher Teil des Textes ersetzt werden muss. Nachdem %-Zeichen steht der **Typbezeichner**: **%d** steht für **d**ecimal (ganzzahlige Ausgabe). Folgende weitere Typen existieren: **%s** für Zeichenketten, **%f** für Fließkommazahlen und **%%** für %.

- Schreibe ein Programm, welches eine Variable mit einem Wert füllt und das Ergebnis auf der Konsole ausgibt.

Lösung:

Lösung 2: Dynamische Ausgabe

```c
#include <stdio.h>
int main(int argc, char **argv)
{
  int iSumOfDigits = 1 + 9 + 8 + 5;
  printf("Some Number: %d\n", iSumOfDigits);
  return 0;
}
```

3. formatierte Ausgabe

Folgende Optionen können die Ausgabe formatieren: **%5d** für mindestens fünf Ziffern, **%05d** für mindestens 5 mit Nullen aufgefüllten Ziffern. Um einen Text linksbündig auszurichten stellt man ein **Minus**-Zeichen voran: `%-5d`. `%-5.2f` hingegen sorgt für 5 Zahlen mindestens, linksbündige und auf zwei Nachkommastellen gerundete Ausgabe.

- Erzeuge folgende Ausgabe, indem du die obigen Informationen benutzt:

```
Mario ist     2m hoch und 00026 Jahre alt
Mario ist 1.8  m hoch und 26    Jahre alt
```

Lösung:

Lösung 3: Formatierung

```c
char pcName[255];
strcpy(pcName,"Mario"); // korrekte Initialisierung
unsigned uiAge = 26u;
```

```c
float fHeight=1.83f;

printf("%s ist %5.0fm hoch und %05d Jahre alt\n", pcName, fHeight, uiAge);
printf("%s ist %-5.1fm hoch und %-05d Jahre alt\n", pcName, fHeight, uiAge);
```

4. Eingabe

Die Eingabe funktioniert analog zur Ausgabe: statt **printf** wird **scanf** benutzt. Wichtig ist dass, bei der Eingabe stets ein **&**-Symbol vor dem Variablennamen geschrieben wird. (Warum wird in einer späteren Veranstaltung erklärt). Für das Einlesen einer Zahl schreibt man zum Beispiel: `scanf("%d",&iZahl);`.

- Schreibe ein Programm, welches dein Alter einliest und es wieder ausgibt. Überprüfe dabei den Rückgabewert, denn dieser muss immer der erfolgreich eingelesenen Variablen entsprechen.
- [Zusatz] Berechne dir die Jahreszahl (gehe dabei von dem Jahr 2011 als feste Größe aus) und gebe das Geburtsjahr auf der Konsole aus.

Lösung:

Lösung 4: Eingabe des Alters

```c
#include <stdio.h><br>
int main(int argc, char **argv)
{
  unsigned int uiAge;
  printf("Alter: "); // Ausgabe vor der Eingabe

  if( 1 != scanf("%d", &uiAge) )
  {
    printf("Konnte das Alter nicht lesen!\n");
    return -1;
  }

  scanf("%d", &uiAge); // &: Adress-Operator, immer mitnehmen!
  printf("Du bist %d Jahre alt.\n", uiAge);
  // Zusatz:
  printf("Du wurdest im Jahre %04d geboren.\n", 2011-uiAge);
  return 0;
}
```

5. Dateiein- und ausgabe

Lese dich mittels der **Manpage** zu **fopen**, **fclose**, **fprintf** sowie **fscanf** in die Grundlagen der Dateiein- und ausgabe ein. (Hinweis: `man 3 fopen`)

- Nenne die drei Schritte die benötigt werden, um eine Datei erfolgreich zu bearbeiten.

  Lösung:

  a) Öffnen
  b) Bearbeiten
  c) Schließen
- Warum sollte man immer **fclose** aufrufen?

  Lösung:

  Sonst kann man innerhalb und ausherhalb(nur unter Windows) des Programmes nicht mehr auf die Datei lesend oder schreibend zugreifen.
- Was ist der Rückgabe wert von **fopen**?

Lösung:

`fopen` liefert einen Dateihandler zurück, der im folgenden die Datei repräsentiert. Ist der Handle `NULL` darf die Datei nicht benutzt werden.

- [Umfangreich] Schreibe ein Programm, welches eine Datei öffnet, die eine Matrix ähnlich des folgenden Beispiels, enthält. Nach dem Lesen soll es die Summe aller Elemente ans Ende der Datei schreiben.

```
1 2 3
4 5 6
7 8 9
```

Lösung:

Lösung 5: Komplexe Ein- und Ausgabe

```c
#include <stdio.h>
int main(int argc, char **argv)
{
  // Schritt 1: Oeffnen
  FILE *pOpenFile = fopen("/tmp/sample.txt", "a+");
  if( NULL == pOpenFile )
  {
    printf("Konnte Datei nicht oeffnen!\n");
    return -1;
  }

  // Schritt 2a: Bearbeiten (lesen)
  int iSum = 0;
  while( feof( pOpenFile ) == 0 )
  {
    int iReadDecimal = 0;
    if( 1 == fscanf(pOpenFile, "%d", &iReadDecimal) )
    {
      iSum+=iReadDecimal;
      printf("Wert gelesen: \"%d\"\nAktuelle Summe: %d\n\n",
          iReadDecimal, iSum);
    }
    else
    {
      printf("Wert konnte nicht gelesen werden ...\n");
    }
  }

  // Schritt 2b: Bearbeiten (schreiben)
  printf("Haenge Ergebnis %d ans Ende der Datei an.\n", iSum)
      ;
  if( 0 == fprintf(pOpenFile, "%d\n", iSum) )
  {
    printf("Inhalt der Datei konnte nicht veraendert werden\n
        ");
  }
  else
  {
    printf("Datei wurde erfolgreich beschrieben!\n");
 }

  // Schritt 3: SchlieBen
  fclose(pOpenFile);
  return 0;
}
```