# Code Examples from Lectures on 3-1_C_basics



### printVar.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/printVar.c)

A simple program demonstrating the use of int variables and printf for ints

```
#include <stdio.h>

int main(void) {
    int answer;
    int result;

    answer = 42;
    result = -99;

    printf("The answer is %d\n", answer);
    printf("The result is %d\n", result);
    return 0;
}
```

### printVar2.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/printVar2.c)

A simple program demonstrating the use of int variables and printf for ints

This does exactly the same thing as printVar.c but is written in a slightly different way

```
#include <stdio.h>

int main(void) {
    //You can declare and initialise variables in one step
    int answer = 42;
    int result;
    result = -99;

    printf("The answer is %d.\nThe result is %d\n", answer,result);

    return 0;
}
```

### printDoubleVar.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/printDoubleVar.c)

A simple demonstration of printing double variables

```
#include <stdio.h>

int main(void) {

    double answer;
    double anotherAnswer = -99;
    answer = 42.83199;
    // %.3lf prints a double variable to 3 decimal places.
    printf("Answers are %.3lf %lf\n",answer,anotherAnswer);

    return 0;
}
```

### scanVar.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/scanVar.c)

A simple program demonstrating the use of scanf

Try running the program and typing illegal input such as hello to see what happens

```
#include <stdio.h>

int main(void) {
    int x;
    double y;

    printf("Enter an int then a double : ");
    scanf("%d %lf",&x,&y);

    printf("You entered %d %lf\n", x,y);
    return 0;
}
```

### expressionExercise.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/expressionExercise.c)

A simple program demonstrating expressions

```
#include <stdio.h>

int main(void){
    int answer1 =  6 * 7 - 8 * 9 /10;  //42 - 72/10 = 42 - 7 = 35
    int answer2 = 2*3*4+5*6;           //6 * 4 + 30 = 24 + 30 = 54
    int answer3 = 5*6/4;               //30/4 = 7
    int answer4 = 3/2;                 //1
    int answer5 = 1/2.0;               //0
    double answer6 = 1/2.0;            //0.50000

    printf("%d\n",answer1);
    printf("%d\n",answer2);
    printf("%d\n",answer3);
    printf("%d\n",answer4);
    printf("%d\n",answer5);
    printf("%lf\n",answer6);

    return 0;
}
```

**modExamples.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/modExamples.c)**

A simple program demonstrating the mod % operator. % gives the remainder

```
#include <stdio.h>

int main(void){


   printf("10 mod 2 is %d\n",10 % 2);       //0
   printf("7 mod 2 is %d\n",7%2);           //1
   printf("2 mod 2 is %d\n",2%2);           //0

   printf("9876 mod 10 is %d\n",9876%10);   //6
   printf("9876 mod 100 is %d\n",9876%100); //76
   printf("9876/10 is %d\n",9876/10);       //987
   return 0;
}
```

**convert.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/convert.c)**

A program to read in an amount of time in hours and convert to minutes

```
#include <stdio.h>
#define MINS_IN_HOURS 60

int main(void) {
    double hours;
    double minutes;

    printf("Please enter the number of hours: ");
    scanf("%lf",&hours);

    minutes = hours * MINS_IN_HOURS;
    printf("That is %.2lf minutes\n",minutes);

    return 0;
}
```

**mystery.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/mystery.c)**

A bad Style version of feet2metres.c

```
#include <stdio.h>


int main(void) {
    double f;
    double m;
    scanf("%lf", &f);
    m = f * 12 * 2.54 / 100;
    printf("%.2lf ", f);
    printf("%.2lf\n", m);
    return 0;
}
```

**feet2metres.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/feet2metres.c)**

Convert a measurement in feet to metres

A simple program demonstrating the use of scanf and #define for constants

```
#include <stdio.h>

#define INCHES_IN_FOOT  12
#define CM_IN_INCH      2.54
#define CM_IN_METRE     100

int main(void) {
    double feet;
    double metres;

    printf("Enter number of feet: ");
    scanf("%lf", &feet);

    metres = feet * INCHES_IN_FOOT * CM_IN_INCH / CM_IN_METRE;

    printf("%.2lf", feet);
    printf(" feet is ");
    printf("%.2lf", metres);
    printf(" metres\n");

    return 0;
}
```

**circle.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/circle.c)**

A simple program to calculate the area of a circle without using the math library

```
#include <stdio.h>

#define PI 3.14159

int main(void) {

    double radius;
    double area;
    printf("Please enter the radius: ");
    scanf("%lf",&radius);
    area = PI * radius * radius;

    printf("The area for radius %lf is %lf\n",radius, area);


    return 0;
}
```

**circleMath.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/circleMath.c)**

A simple program to calculate the area of a circle using the math library

```
If you are using gcc you will need to compile with the -lm flag eg
gcc -Wall -Werror -O -o circleMath circleMath.c -lm
```

```
#include <stdio.h>
#include <math.h>



int main(void) {

    double radius;
    double area;
    printf("Please enter the radius: ");
    scanf("%lf",&radius);
    //M_PI is a constant from the math library
    //pow is a function from the maths library
    //(we are using it to do radius to the power of 2)
    area = M_PI * pow(radius,2);

    printf("The area for radius %lf is %lf\n",radius, area);


    return 0;
}
```

**sum2a.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/sum2a.c)**

A simple program demonstrating the use of scanf to sum 2 numbers

```
#include <stdio.h>

int main(void) {
    int x, y, sum;
    printf("Enter x: ");
    scanf("%d", &x);
    printf("Enter y: ");
    scanf("%d", &y);
    sum = x + y;
    // These 6 printfs can be better replaced by a single printf
    printf("%d", x);
    printf(" + ");
    printf("%d", y);
    printf(" = ");
    printf("%d", sum);
    printf("\n");
    return 0;
}
```

**sum2b.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/sum2b.c)**

A simple program demonstrating the use of scanf to sum 2 numbers

```
#include <stdio.h>

int main(void) {
    int x, y, sum;
    printf("Enter x: ");
    scanf("%d", &x);
    printf("Enter y: ");
    scanf("%d", &y);
    sum = x + y;
    printf("%d + %d = %d\n", x, y, sum);
    return 0;
}
```

**sum2c.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/sum2c.c)**

A simple program demonstrating the use of scanf to sum 2 numbers

```
#include <stdio.h>

int main(void) {
    int x, y;
    printf("Enter x: ");
    scanf("%d", &x);
    printf("Enter y: ");
    scanf("%d", &y);
    printf("%d + %d = %d\n", x, y, x + y);
    return 0;
}
```

**arithGotcha.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/arithGotcha.c)**

A simple program demonstrating integer division and divide by zero issues that can trip students up

```
#include <stdio.h>

int main(void){
     int answer1;
     int answer2;
     answer1 = 1/2;   // int division so answer1 = 0

     printf("1/2  is %d\n",answer1);
     answer2 = answer1/answer1;  // 0/0 is undefined
     printf("1/2 divided by 1/2 is %d\n",answer2);

     return 0;
}
```

**arithGotchaFixed.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/arithGotchaFixed.c)**

A simple program demonstrating fixing issues from arithGotcha.c

```c
#include <stdio.h>

int main(void){
     double answer1;
     double answer2;
     answer1 = 1.0/2;   // double division so answer1 is 0.5

     printf("1/2  is %lf\n",answer1);
     answer2 = answer1/answer1;
     printf("1/2 divided by 1/2 is %lf\n",answer2);

     return 0;
}
```

**defineBeware.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/defineBeware.c)**

An example to show you the dangers of defining expressions with #define. In this subject just stick to constant values

Angela Finlayson angf@cse.usw.edu.au

To see what C does with #define and #include run gcc -E defineBeware.c

```c
#include <stdio.h>

#define NUM1 30
#define NUM2 40

#define SUM NUM1+NUM2
#define AVERAGE  SUM/2


int main(void){
    printf("Average is %d\n",AVERAGE);
    return 0;
}
```

**defineBewareFixed.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/defineBewareFixed.c)**

An example to show you the dangers of defining expressions with #define. In this subject just stick to constant values

Angela Finlayson angf@cse.usw.edu.au

To see what C does with #define and #include run gcc -E defineBeware.c

```c
#include <stdio.h>

#define NUM1 30
#define NUM2 40

//If you are going to do this kind of thing
//make sure you use brackets
#define SUM (NUM1+NUM2)
#define AVERAGE  (SUM/2)


int main(void){

    printf("Average is %d\n",AVERAGE);
    return 0;
}
```

**percentage.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/3-1_C_basics/code/percentage.c)**

Printing a % character

Author : Angela Finlayson

This is a basic template for any C program you will write

```c
#include <stdio.h>

int main(void){

    //To print one % you need 2 % characters
    printf("I got 100%%\n");

    return 0;
}
```