

ESC101N

Fundamentals of Computing

Arnab Bhattacharya
arnabb@iitk.ac.in

Indian Institute of Technology, Kanpur
<http://www.iitk.ac.in/esc101/>

1st semester, 2010-11
Tue, Wed, Fri 0800-0900 at L7

Program to add two numbers

```
#include <stdio.h>

int main()
{
    int a, b, c;

    scanf("%d", &a);
    scanf("%d", &b);

    c = a + b;

    printf("%d\n", c);
}
```

- Save it as the file `addition.c`
- Compile using `gcc addition.c` and then run it by `./a.out`
- Play with the program to understand it
- Introduce errors to see what happens

A little more verbose version

```
/* Program to add two numbers */

#include <stdio.h> // Include headers

int main() // Main function
{
    int a, b, c; // Declare variables

    scanf("%d", &a); // Read 'a' from keyboard
    scanf("%d", &b); // Read 'b' from keyboard

    c = a + b;

    printf("%d\n", c); // Write 'c' to screen
}
```

Crash course in C

- Comments in a program can be in two ways
 - They are not translated into machine language by compiler – gcc simply ignores them

```
/* Program ... */  
#include ... // Include header
```

Crash course in C

- Comments in a program can be in two ways
 - They are not translated into machine language by compiler – gcc simply ignores them

```
/* Program ... */  
#include ... // Include header
```

- In general, certain other standard files are needed
 - It is best to ignore the exact meaning now

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

Crash course in C

- Comments in a program can be in two ways
 - They are not translated into machine language by compiler – gcc simply ignores them

```
/* Program ... */  
#include ... // Include header
```

- In general, certain other standard files are needed
 - It is best to ignore the exact meaning now

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

- Actual execution of the program begins at `main`
 - Every instruction inside `main` is executed step by step

```
int main()  
{  
...  
}
```

Crash course in C (contd.)

- Variables help define the components of a program
 - a, b, c represent the different numbers
- Types of each variable must be defined
 - a, b, c are integers

```
int a, b, c;
```

Crash course in C (contd.)

- Variables help define the components of a program
 - a, b, c represent the different numbers
- Types of each variable must be defined
 - a, b, c are integers

```
int a, b, c;
```

- Reading (or input of) a number is done using `scanf`
 - Ignore the exact syntax for now

```
scanf("%d", &a);  
scanf("%d", &b);
```


Crash course in C (contd.)

- Variables help define the components of a program
 - a, b, c represent the different numbers
- Types of each variable must be defined
 - a, b, c are integers

```
int a, b, c;
```

- Reading (or input of) a number is done using `scanf`
 - Ignore the exact syntax for now

```
scanf("%d", &a);  
scanf("%d", &b);
```

- Actual arithmetic (or logical) operations are written using operators
 - Addition is denoted by +

```
c = a + b;
```

Crash course in C (contd.)

- Variables help define the components of a program
 - a, b, c represent the different numbers
- Types of each variable must be defined
 - a, b, c are integers

```
int a, b, c;
```

- Reading (or input of) a number is done using `scanf`
 - Ignore the exact syntax for now

```
scanf("%d", &a);  
scanf("%d", &b);
```

- Actual arithmetic (or logical) operations are written using operators
 - Addition is denoted by +

```
c = a + b;
```

- Writing (or output of) a number is done using `printf`
 - Ignore the exact syntax for now

```
printf("%d\n", c);
```