

## Printing

```
printf("Hello\n");
printf("World");
printf("!\n");
```

## Variables and Data Types

```
/*
Names are case-sensitive and may begin with:
    letters, _
After, may include
    letters, numbers, _
Convention says
    Start with a lowercase word, then additional words are capitalized
    ex. myFirstVariable
*/
char testGrade = 'A'; // single 8-bit character.
char name[] = "Mike"; // array of characters (string)

// you can make them unsigned by adding "unsigned" prefix
short age0 = 10;      // at least 16-bits signed integer
int age1 = 20;        // at least 16-bits signed integer (not smaller than short)
long age2 = 30;       // at least 32-bits signed integer
long long age3 = 40;  // at least 64-bits signed integer

float gpa0 = 2.5;     // single precision floating point
double gpa1 = 3.5;    // double-precision floating point
long double gpa2 = 3.5; // extended-precision floating point

int isTall;          // 0 if false, non-zero if true
isTall = 1;

testGrade = 'F';

printf("%s, your grade is %c \n", name, testGrade);
/*
%c    character
%d    integer number (base 10)
%e    exponential floating-point number
%f    floating-point number
%i    integer (base 10)
%o    octal number (base 8)
%s    a string of characters
%u    unsigned decimal (integer) number
*/
```

`%x`    *number in hexadecimal (base 16)*  
`%%`    *print a percent sign*  
`\%`    *print a percent sign*  
`*/`

## Casting and Converting

```
printf("%d \n", (int)3.14);  
printf("%f \n", (double)3 / 2);
```

## Pointers

```
int num = 10;  
printf("%p \n", &num);
```

```
int *pNum = &num;  
printf("%p \n", pNum);  
printf("%d \n", *pNum);
```

## Numbers

```
printf("%d \n", 2 * 3);    // Basic Arithmetic: +, -, /, *  
printf("%d \n", 10 % 3);    // Modulus Op. : returns remainder of 10/3  
printf("%d \n", 1 + 2 * 3); // order of operations  
printf("%f \n", 10 / 3.0); // int's and doubles
```

```
int num = 10;  
num += 100;                // +=, -=, /=, *=  
printf("%d \n", num);
```

```
num++;  
printf("%d \n", num);
```

```
printf("%f \n", pow(2, 3));  
printf("%f \n", sqrt(144));  
printf("%f \n", round(2.7));
```

## User Input

```
char name[10];
printf("Enter your name: ");
fgets(name, 10, stdin);
printf("Hello %s! \n", name);

int age;
printf("Enter your age: ");
scanf("%d", &age);
printf("You are %d \n", age);

char grade;
printf("Enter your grade: ");
scanf("%c", &grade);
printf("You got an %c on the test \n", grade);

double gpa;
printf("Enter your gpa: ");
scanf("%lf", &gpa);
printf("Your gpa is %f \n", gpa);
```

## Arrays

```
// int luckyNumbers[6];
int luckyNumbers[] = {4, 8, 15, 16, 23, 42};
//   indexes:  0 1 2 3 4 5
luckyNumbers[0] = 90;
printf("%d \n", luckyNumbers[0]);
printf("%d \n", luckyNumbers[1]);
```

## 2 Dimensional Arrays

```
// int numberGrid[2][3];
int numberGrid[2][3] = { {1, 2, 3}, {4, 5, 6} };
numberGrid[0][1] = 99;

printf("%d \n", numberGrid[0][0]);
printf("%d \n", numberGrid[0][1]);
```

## Functions

```
int addNumbers(int num1, int num2);

int main(){
    int sum = addNumbers(4, 60);
    printf("%d \n", sum);

    return 0;
}

int addNumbers(int num1, int num2){
    return num1 + num2;
}
```

## If Statements

```
int isStudent = 0;
int isSmart = 0;

if(isStudent != 0 && isSmart != 0){
    printf("You are a student\n");
} else if(isStudent != 0 && isSmart == 0){
    printf("You are not a smart student\n");
} else {
    printf("You are not a student and not smart\n");
}

// >, <, >=, <=, !=, ==
if(1 > 3){
    printf("number omparison was true\n");
}

if('a' > 'b'){
    printf("character comparison was true\n");
}
```

## Switch Statements

```
char myGrade = 'A';
switch(myGrade){
    case 'A':
        printf("You Pass\n");
        break;
    case 'F':
        printf("You fail\n");
        break;
    default:
        printf("Invalid grade\n");
}
```

## While Loops

```
int index = 1;
while(index <= 5){
    printf("%d \n", index);
    index++;
}
```

```
index = 1;
do{
    printf("%d \n", index);
    index++;
}while(index <= 5);
```

## For Loops

```
for(int i = 0; i < 5; i++){
    printf("%d \n", i);
}
```

## Structs

```
struct Book{
    char title[100];
    char author[100];
}

int main(){

    struct Book book1;
    book1.numPages = 600;
    strcpy( book1.title, "Harry Potter" );
    strcpy( book1.author, "JK Rowling");

    printf("%s \n", book1.title);

    return 0;
}
```