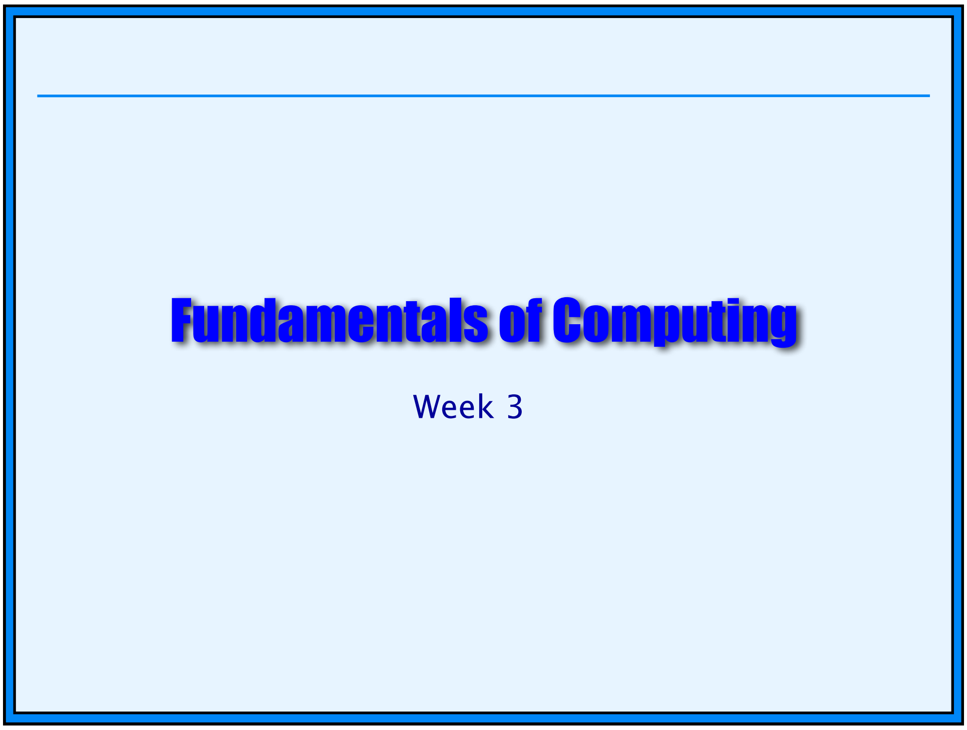
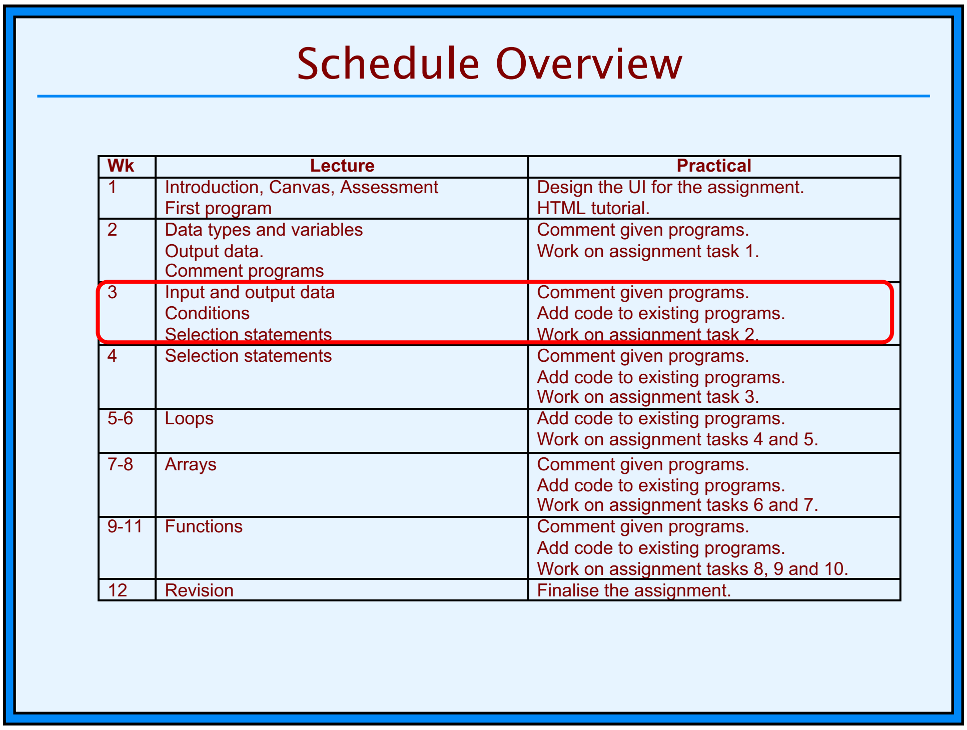


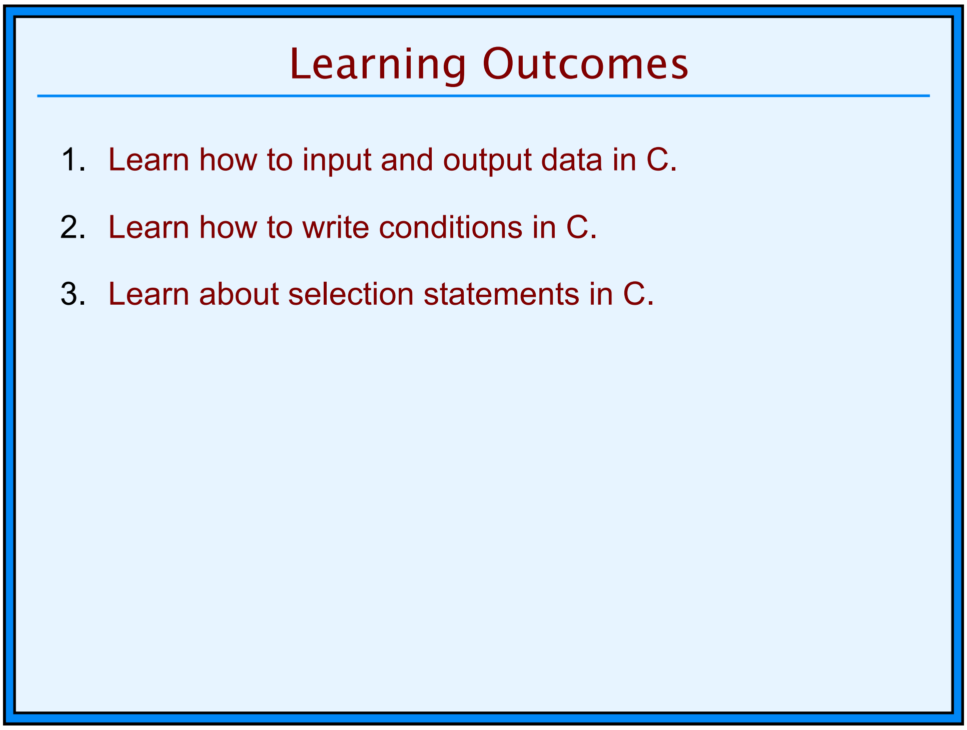
Fundamentals of Computing

Week 3



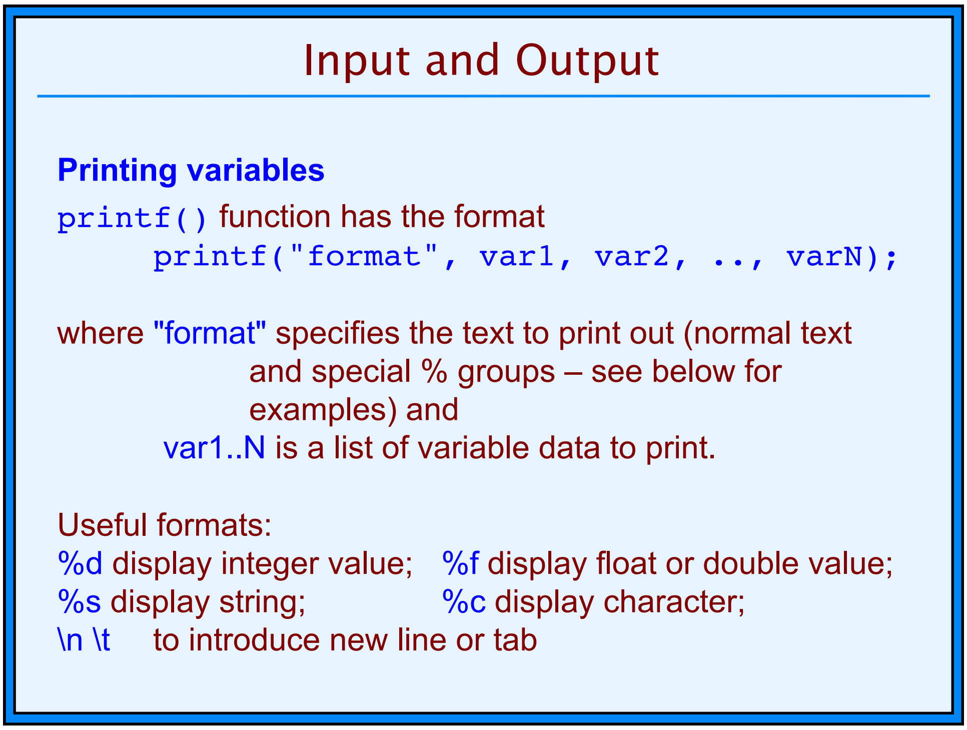
Schedule Overview

Wk	Lecture	Practical
1	Introduction, Canvas, Assessment First program	Design the UI for the assignment. HTML tutorial.
2	Data types and variables Output data. Comment programs	Comment given programs. Work on assignment task 1.
3	Input and output data Conditions Selection statements	Comment given programs. Add code to existing programs. Work on assignment task 2.
4	Selection statements	Comment given programs. Add code to existing programs. Work on assignment task 3.
5-6	Loops	Add code to existing programs. Work on assignment tasks 4 and 5.
7-8	Arrays	Comment given programs. Add code to existing programs. Work on assignment tasks 6 and 7.
9-11	Functions	Comment given programs. Add code to existing programs. Work on assignment tasks 8, 9 and 10.
12	Revision	Finalise the assignment.



Learning Outcomes

1. Learn how to input and output data in C.
2. Learn how to write conditions in C.
3. Learn about selection statements in C.



Input and Output

Printing variables

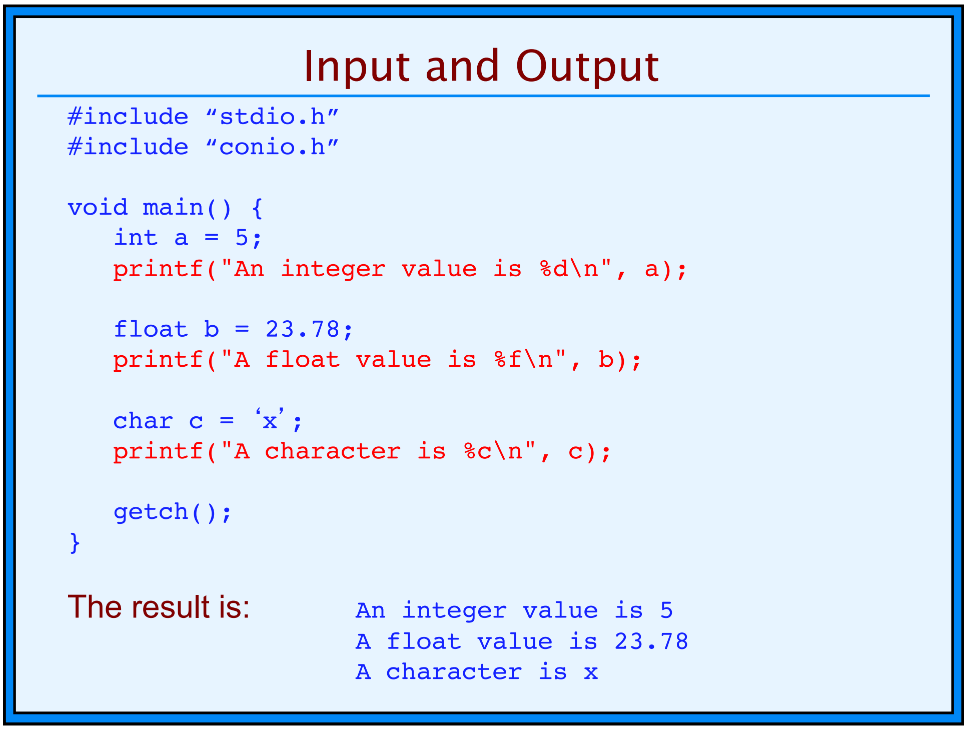
`printf()` function has the format

```
      printf("format", var1, var2, .., varN);
```

where "format" specifies the text to print out (normal text and special % groups – see below for examples) and
var1..N is a list of variable data to print.

Useful formats:

%d display integer value; %f display float or double value;
%s display string; %c display character;
\n \t to introduce new line or tab

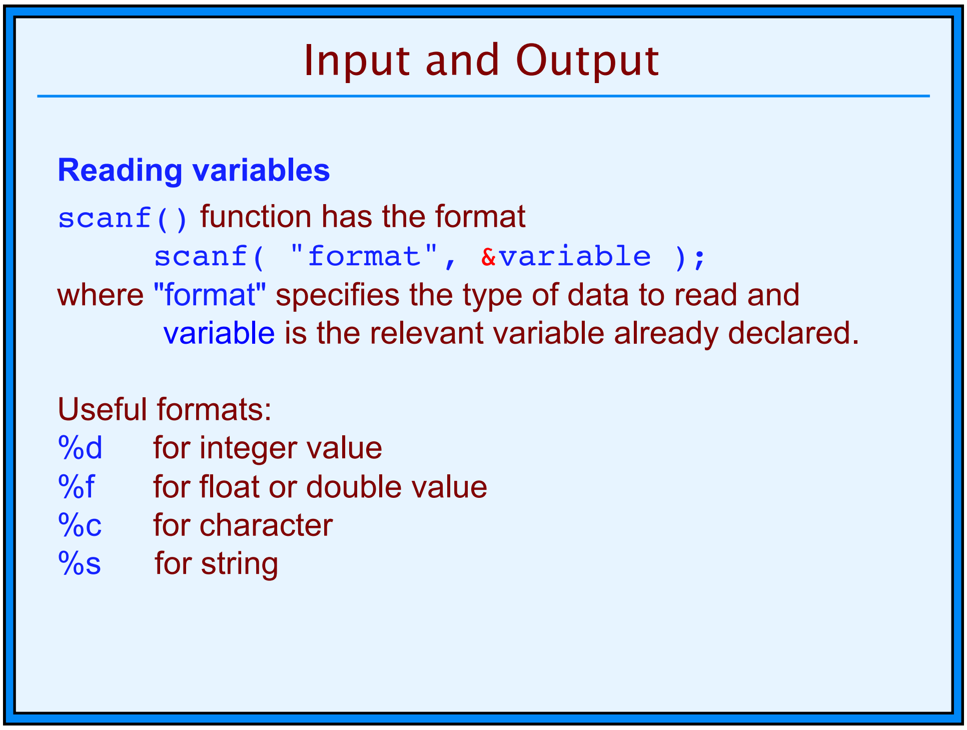


Input and Output

```
#include “stdio.h”
#include “conio.h”

void main() {
   int a = 5;
   printf("An integer value is %d\n", a);

   float b = 23.78;
   printf("A float value is %f\n", b);

   char c = ‘x’;
   printf("A character is %c\n", c);

   getch();
}
```

The result is:

```
An integer value is 5
A float value is 23.78
A character is x
```



Input and Output

Reading variables

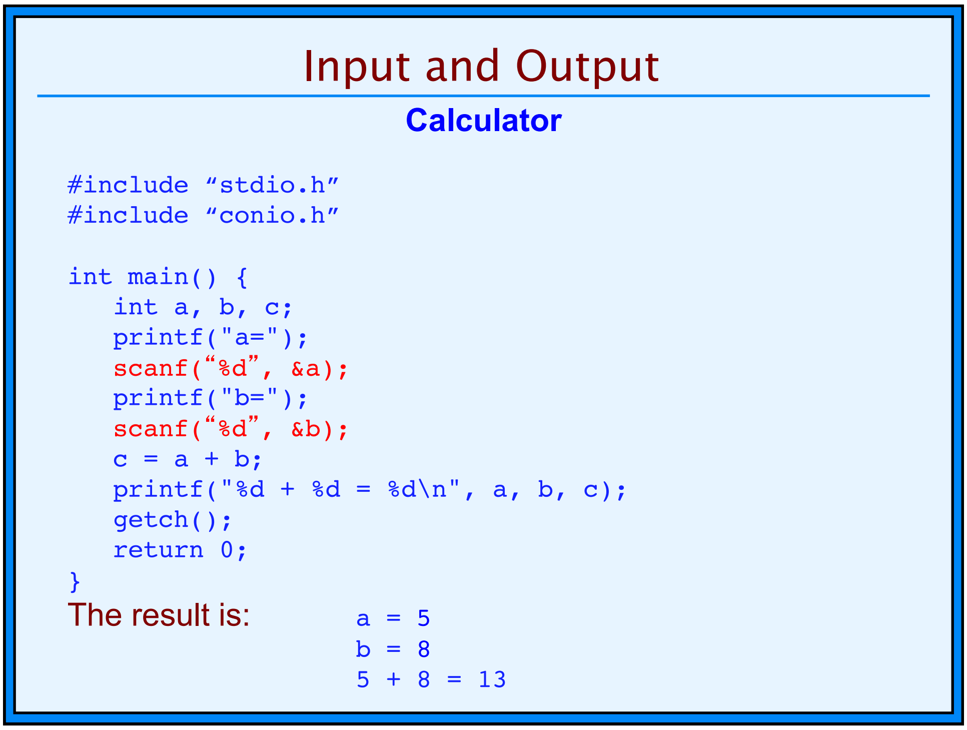
`scanf()` function has the format

```
scanf( "format", &variable );
```

where "format" specifies the type of data to read and
variable is the relevant variable already declared.

Useful formats:

%d	for integer value
%f	for float or double value
%c	for character
%s	for string



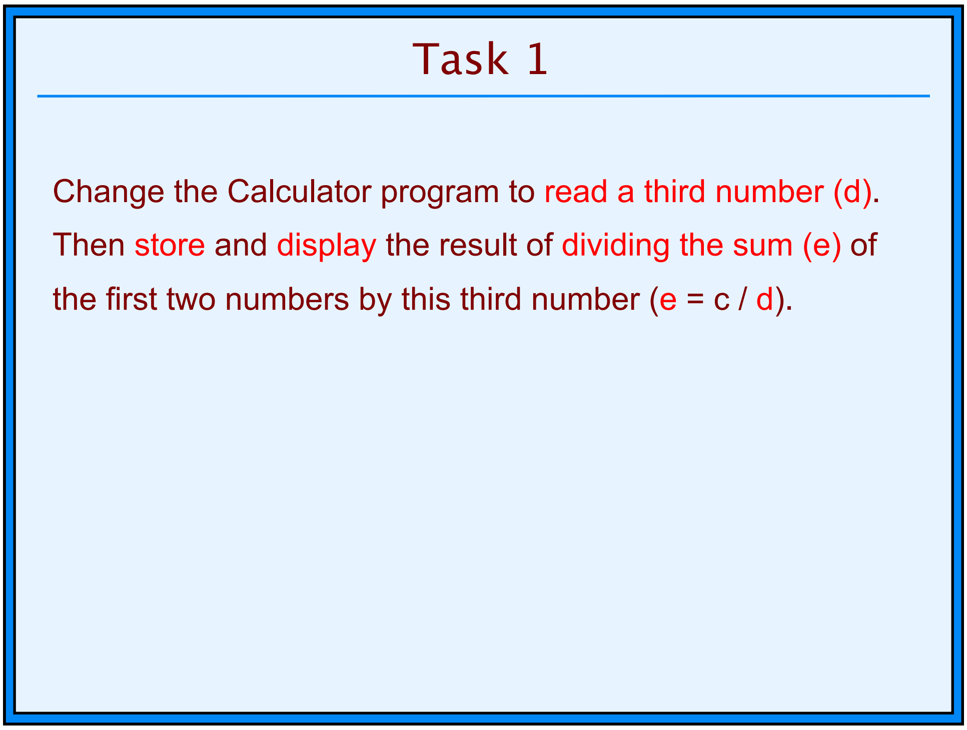
Input and Output

Calculator

```
#include “stdio.h”
#include “conio.h”

int main() {
   int a, b, c;
   printf("a=");
   scanf(“%d”, &a);
   printf("b=");
   scanf(“%d”, &b);
   c = a + b;
   printf("%d + %d = %d\n", a, b, c);
   getch();
   return 0;
}
```

The result is:

```
a = 5
b = 8
5 + 8 = 13
```



Task 1

Change the Calculator program to read a third number (d). Then store and display the result of dividing the sum (e) of the first two numbers by this third number (e = c / d).

Conditions

Conditions are logical operators involving the comparison of quantities.

Condition	Meaning
a < b	a less than b
a <= b	a less than or equal to b
a > b	a greater than b
a >= b	a greater than or equal to b
a == b	a equals b
a != b	a not equal to b

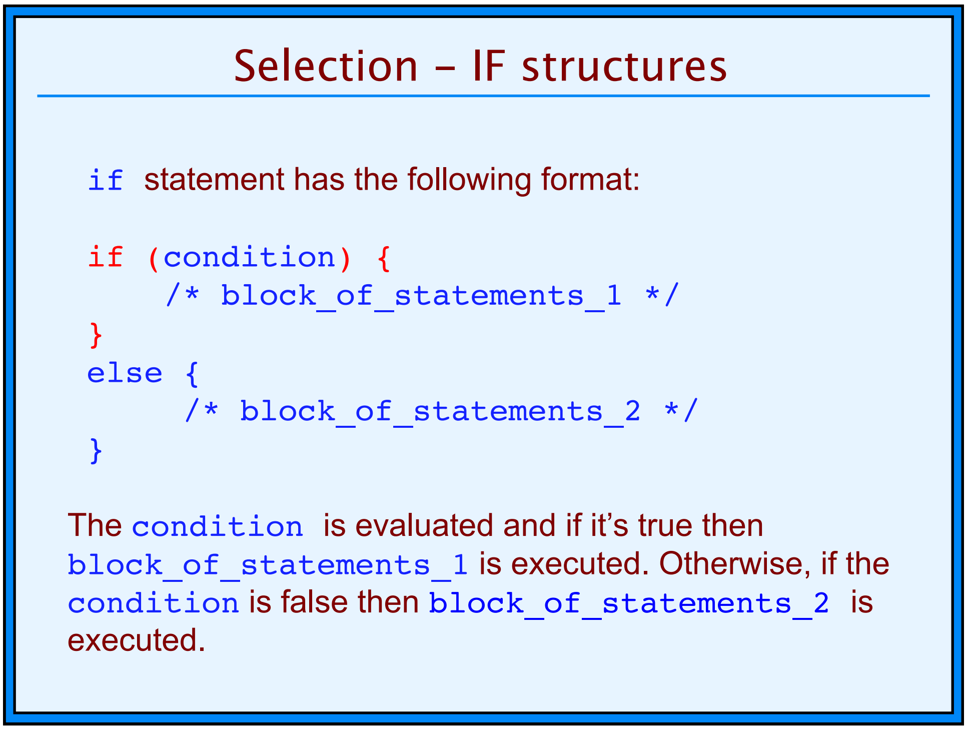
Conditions

The boolean operators are:

Boolean operators	Meaning
&&	and
||	or
!	not

The boolean operators are for combining more than one condition.

!(a < b && a > c) == (a > b || a < c)

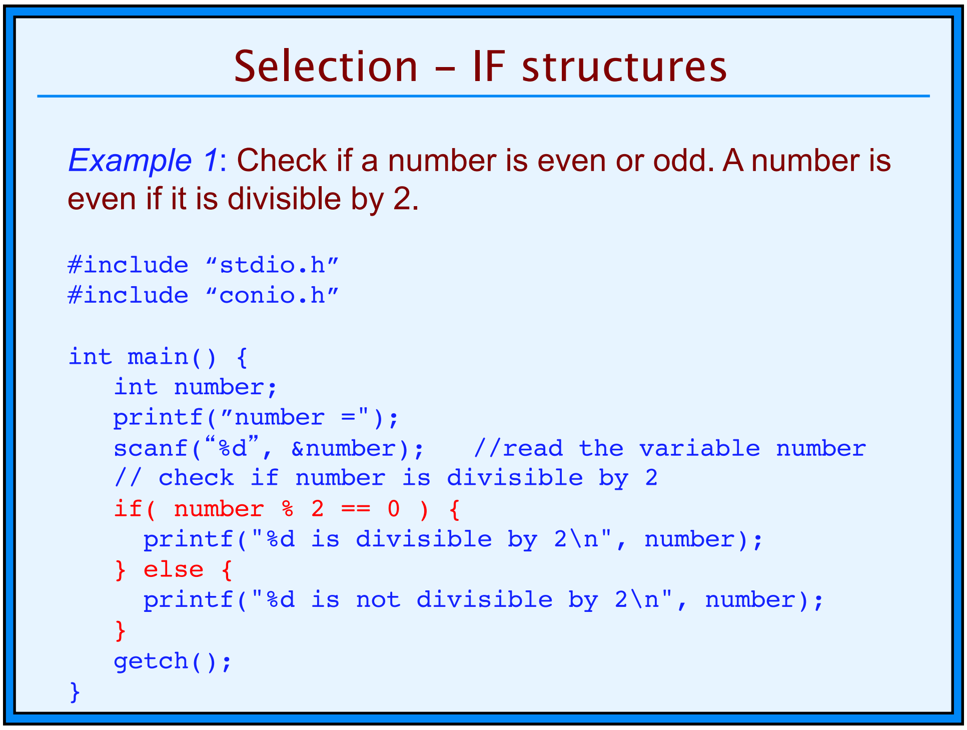


Selection – IF structures

if statement has the following format:

```
if (condition) {
    /* block_of_statements_1 */
}
else {
     /* block_of_statements_2 */
}
```

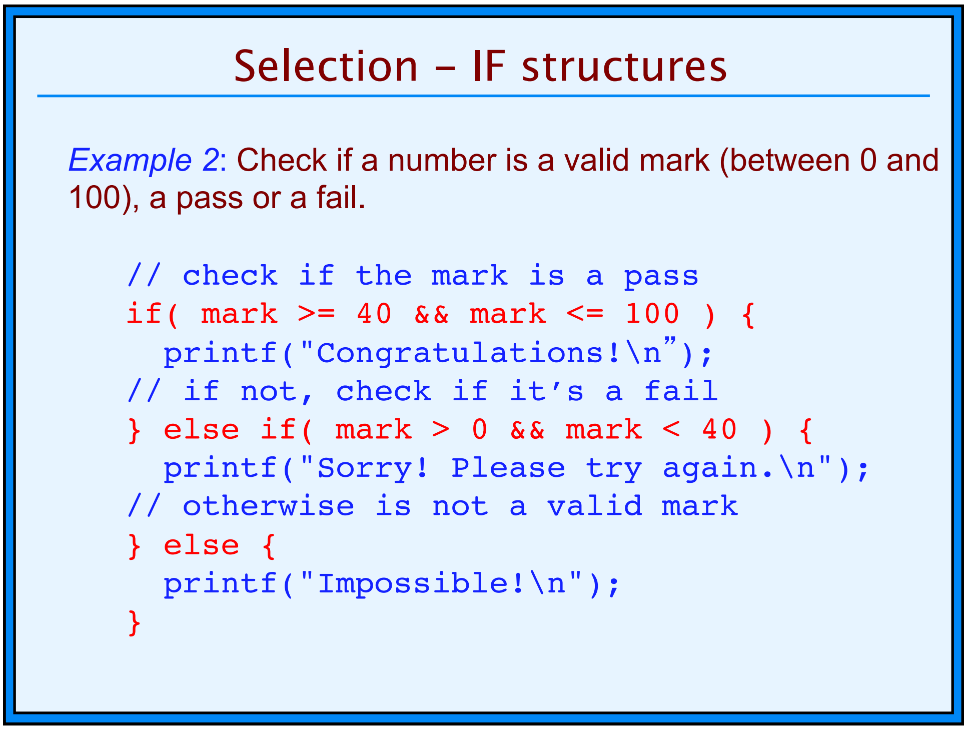
The condition is evaluated and if it’s true then block_of_statements_1 is executed. Otherwise, if the condition is false then block_of_statements_2 is executed.



Selection – IF structures

Example 1: Check if a number is even or odd. A number is even if it is divisible by 2.

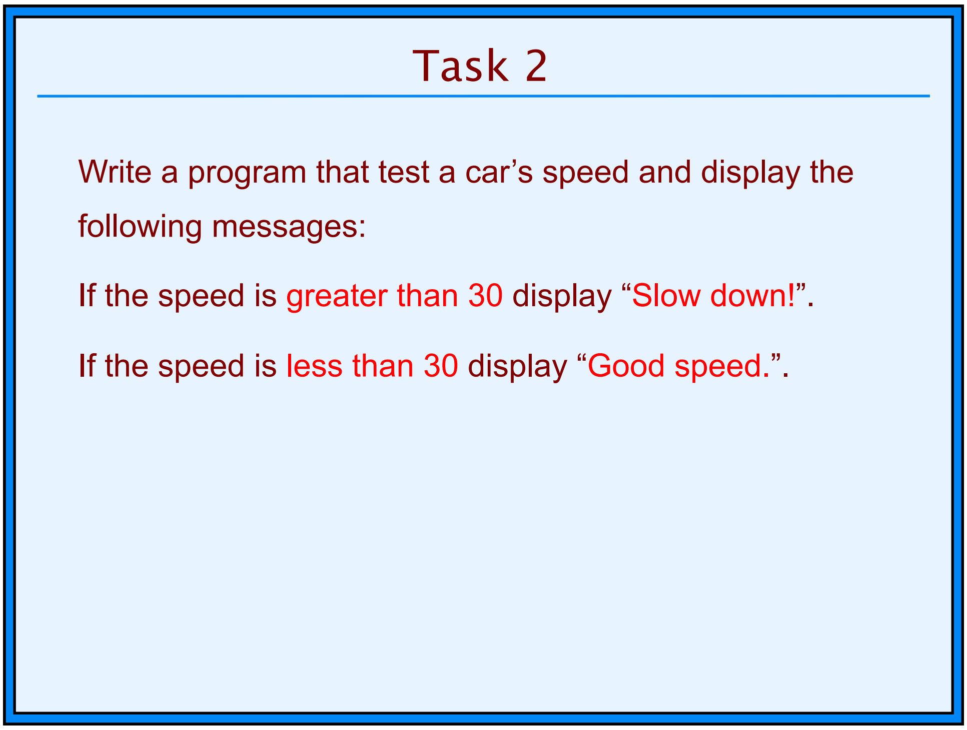
```
#include “stdio.h”
#include “conio.h”

int main() {
   int number;
   printf(”number =");
   scanf(“%d”, &number);   //read the variable number
   // check if number is divisible by 2
   if( number % 2 == 0 ) {
     printf("%d is divisible by 2\n", number);
   } else {
     printf("%d is not divisible by 2\n", number);
   }
   getch();
}
```



Selection – IF structures

Example 2: Check if a number is a valid mark (between 0 and 100), a pass or a fail.

```
// check if the mark is a pass
if( mark >= 40 && mark <= 100 ) {
  printf("Congratulations!\n”);
// if not, check if it’s a fail
} else if( mark > 0 && mark < 40 ) {
  printf("Sorry! Please try again.\n");
// otherwise is not a valid mark
} else {
  printf("Impossible!\n");
}
```

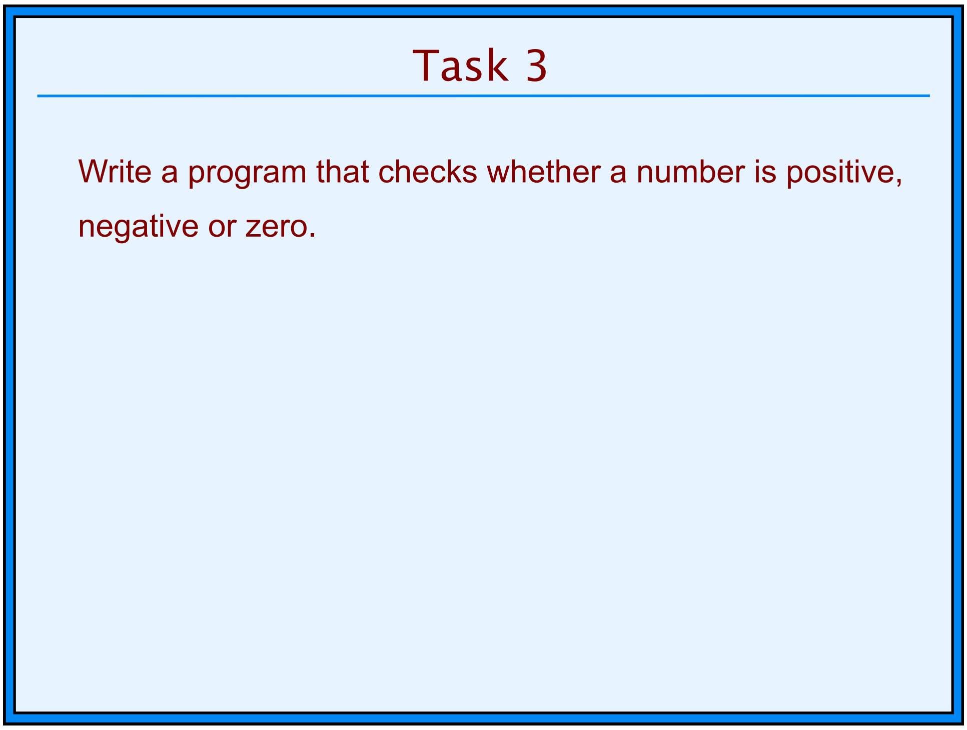


Task 2

Write a program that test a car’s speed and display the following messages:

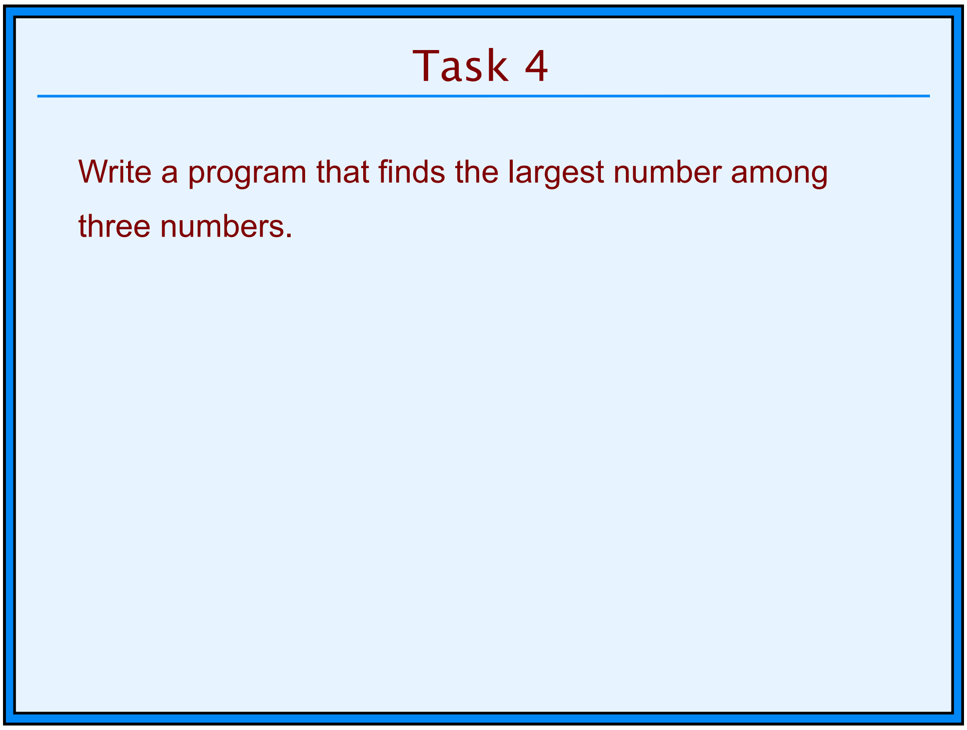
If the speed is greater than 30 display “Slow down!”.

If the speed is less than 30 display “Good speed.”.



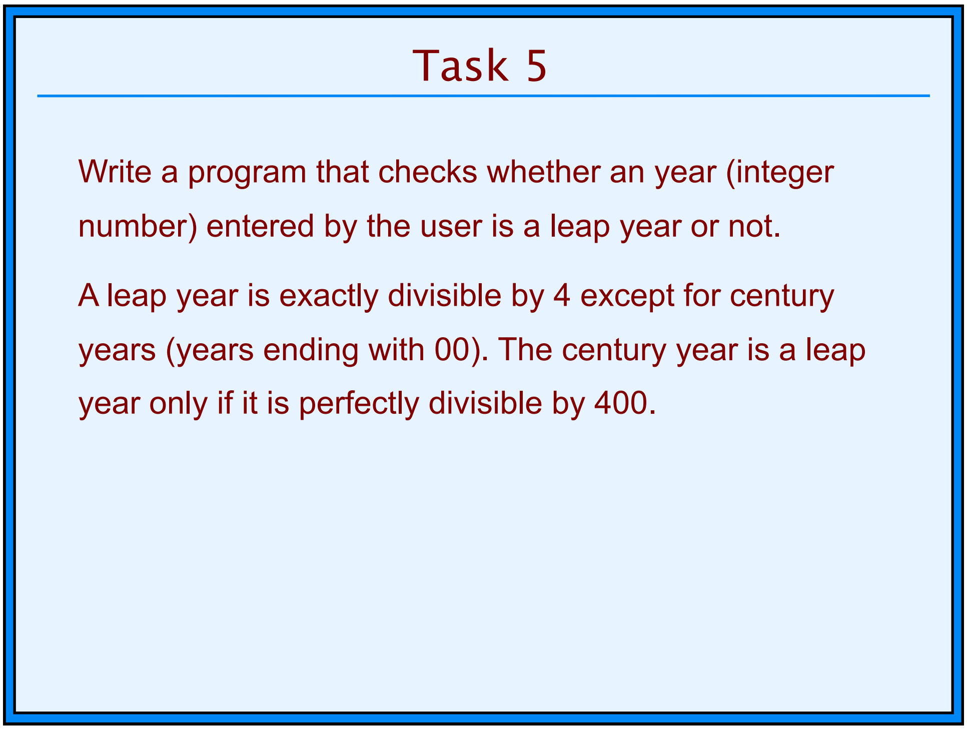
Task 3

Write a program that checks whether a number is positive, negative or zero.



Task 4

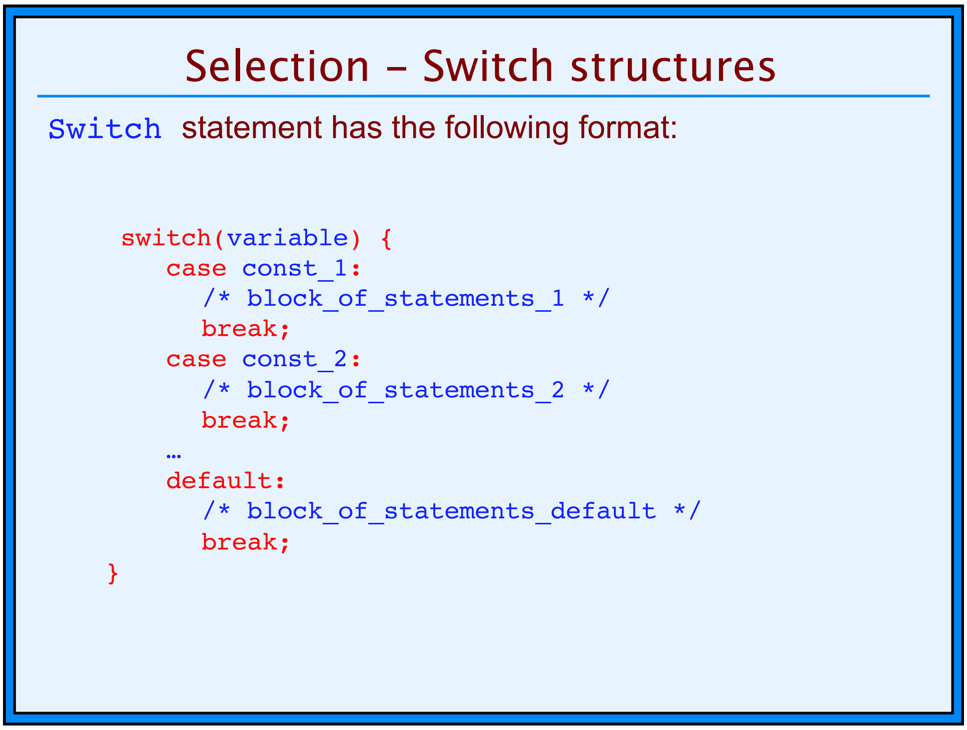
Write a program that finds the largest number among three numbers.



Task 5

Write a program that checks whether an year (integer number) entered by the user is a leap year or not.

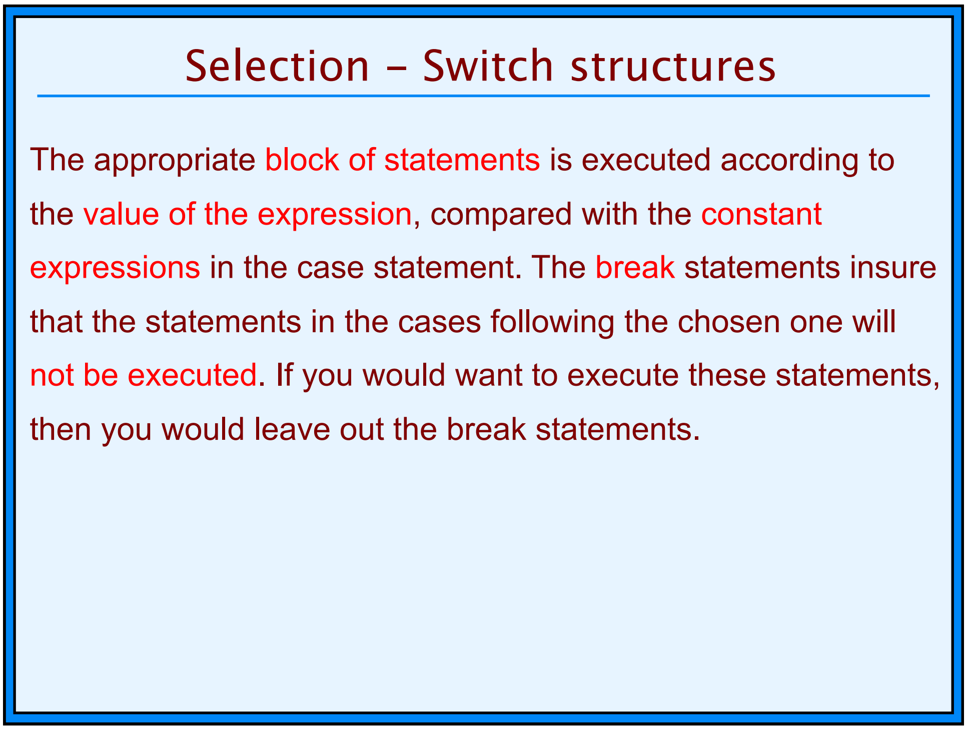
A leap year is exactly divisible by 4 except for century years (years ending with 00). The century year is a leap year only if it is perfectly divisible by 400.



Selection – Switch structures

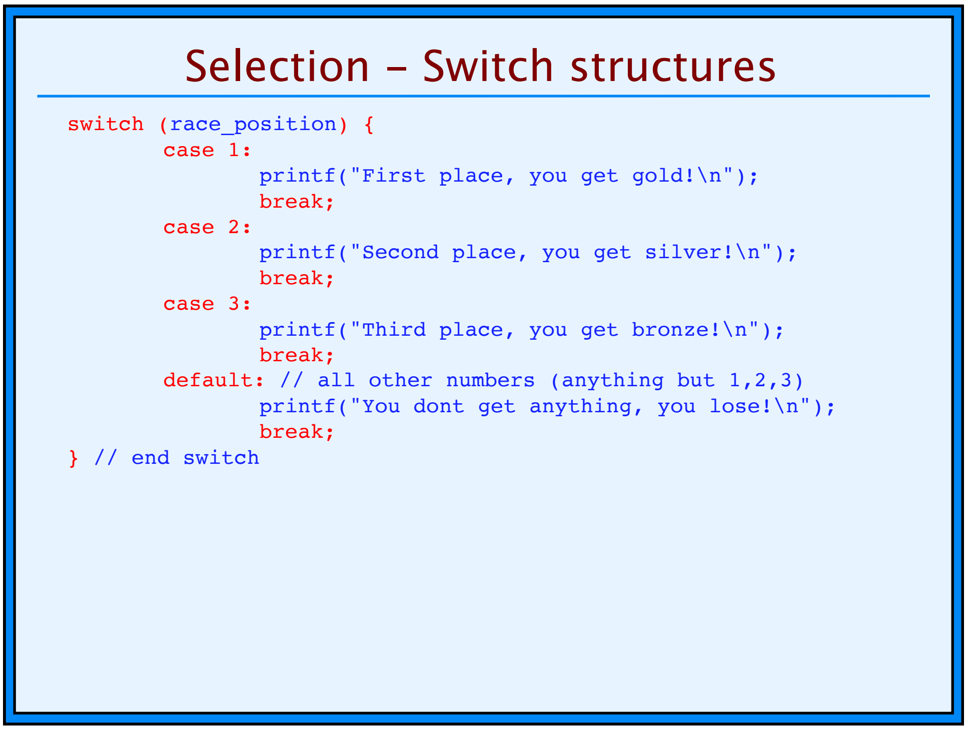
`Switch` statement has the following format:

```
switch(variable) {
   case const_1:
      /* block_of_statements_1 */
      break;
   case const_2:
      /* block_of_statements_2 */
      break;
   …
   default:
      /* block_of_statements_default */
      break;
}
```



Selection – Switch structures

The appropriate block of statements is executed according to the value of the expression, compared with the constant expressions in the case statement. The break statements insure that the statements in the cases following the chosen one will not be executed. If you would want to execute these statements, then you would leave out the break statements.



Selection – Switch structures

```
switch (race_position) {
        case 1:
                printf("First place, you get gold!\n");
                break;
        case 2:
                printf("Second place, you get silver!\n");
                break;
        case 3:
                printf("Third place, you get bronze!\n");
                break;
        default: // all other numbers (anything but 1,2,3)
                printf("You dont get anything, you lose!\n");
                break;
} // end switch
```