Fundamentals of Computing: Lecture 4

Piyush P Kurur
Office no: 224
Dept. of Comp. Sci. and Engg.
IIT Kanpur

August 3, 2009

Summary of previous lecture (Variables)

- Variables are memory location where values are stored
- They have a name, a type associated with them and a value.
- The name of a variable can start with a letter and contain letter or digit.
- The special character _ (under score) is considered as a letter.
- The value associated can be changed using an assignment.

Summary Operator precedence

- Arithmetic operators

- relational operators
- boolean operators

Summary Operator precedence

- Arithmetic operators
  - Unary operators (unary -)
  - *, /
  - +, −
- relational operators
- boolean operators

Summary Operator precedence

- Arithmetic operators
  - Unary operators (unary -)
  - `*`, /
  - `+`, `-`
- relational operators
- boolean operators

eg `- 4 * 3 < 1 && 2 > x + 5` is same as
`((-4) * 3) < 1) && (2 > (x + 5))`

Integer expressions

- Variable declaration

```
        int x;
        int foo=100;
```

- Printing

```
        printf("The value of integer variable %d\n",x);
```

- Arithmetic operators +,−,∗,/, %, − (unary minus) etc
- Relational operators. <,<=,>, >=, == etc

Important

The operator for checking for equality is == and not =.

Factorial program

```
# include<stdio.h>

int main(){
  int n;
  int i = 1;
  printf("Enter the value: ");
  scanf("%d",&n);
  int fact = 1;
  while(i <= n)
  {
    fact = fact * i;
    i = i + 1;
  }
  printf("The factorial of %d is %d\n", n, fact);
}
```

Why did the factorial program go wrong?

Why did the factorial program go wrong?

Answer

Integers are of fixed precision typically 32 bits.

Real numbers expressions

- Variable declaration

```
float x;
float pi=3.141;
double avagadro = 6.023e23;
```

- Printing

```
printf("Values are %f, %g\n",x,avagadro);
```

- Arithmetic operators and relational operators are similar to integers.

Important

Use double always. That gives better precission.

Integers and Floats

C does automatic coversion between integers and floats.

- Integer to Float/Double extension
- Float/Double to integer truncation

Unfortunately this is a very bad design.

```
int u = 10;
int v = 11;
float av;
av = (u + v)/2
prinf("%f",av);
```

Assignments

Assignment is used to modify the value of a variable.
eg.

```
x = 10;
foo = 4.2;
```

Assignments

Assignment is used to modify the value of a variable.
eg.

```
x = 10;
foo = 4.2;
```

Assignment as expression

In C assignment itself is an expression.

```
  x = y = 10;
```

Assignments

Assignment is used to modify the value of a variable.
eg.

```
x = 10;
foo = 4.2;
```

Assignment as expression

In C assignment itself is an expression.

```
x = y = 10;
```

Special assignment

```
i++;              i = i + 1;
foo *= 10;        foo = foo * 10;
```

Boolean

There are no booleans in C.
Integers, characters etc all play the role of boolean
value of 0 is false. value of nonzero is true.

Boolean

There are no booleans in C.
Integers, characters etc all play the role of boolean
value of 0 is false. value of nonzero is true.

WARNING

```
x = 100;
if (x = 0)
{
  printf("Null value unexpected");
}else{
  printf("Good value");
}
```