

Tutorial Sheet (August 24, 2018)

ESC101 – Fundamentals of Computing

Announcements

1. **Extra lab for B10, 11, 12, 14**, August 25, 2018, CC-01, CC-02, 2PM
 2. **Extra session** for students facing difficulty with English lectures but who can understand Hindi, August 25, 2018, CC-02, 5PM
 3. List of course TA/tutor/admin who speak other Indian languages
web.cse.iitk.ac.in/users/purushot/courses/esc/2018-19-a/material/language.pdf
 4. Course schedule with list of extra classes, labs, holidays
web.cse.iitk.ac.in/users/purushot/courses/esc/2018-19-a/material/schedule.pdf
 5. **Major Quiz**, August 29, 2018 12PM, L20
 - a. Bring your **institute ID card, pencil, eraser, sharpener**
 - b. Answers to be written on the question paper itself
 6. Students are warned that any use of mobile phones, iPads, laptops or other electronic communication devices during labs, exams, quizzes will lead to the student being immediately asked to leave, as well as possibly given a zero for the whole lab/exam.
-

Demonstrations

1. Please demonstrate to students how to see their Grade Card on Prutor, as well as go to CodeBook to see their scores, remarks on their scores, as well as make regrading requests.
 2. Please show students how autograding was done (lec9 slide 3)
-

Revision (ask for doubts)

1. **Increment, decrement operators:** both `++a` and `a++` increment value of `a` and store incremented value in `a`. But `a++` generates the old value whereas `++a` generates new value. Can use with `int`, `long`, `float`, `double`.
2. `++3` or `3++` will cause errors. Mr. C will try `3 = 3 + 1` which is wrong.

3. **Compound assignment** is useful `a += 5`, `b *= c`;
 4. **Relational Operators**: `<`, `<=`, `==`, `!=`, `>=`, `>`. Be careful about their precedence (write down BODMAS table in notebook). All relational expressions generate 0 (interpreted as FALSE) or 1 (interpreted as TRUE).
 5. **Logical Operators**: `&&`, `||`, and `!` are used to construct powerful conditional expressions. Be careful to use brackets.
 6. **if, if-else Statements**: Careful to put curly brackets, especially if multiple statements inside if part or else part. Indentation is ignored by Mr. C. Use brackets to tell him what all to do in if/else.
 7. **Ternary Operator**: cute shortcut, do not misuse/overuse.
 8. **Switch Statement**: a shortcut that only checks equality and that too only for int. For inequality checks, float etc, use if-else yourself. Be careful about fall-through and use of break, default.
-

Sample Questions to discuss

Some cute ways to manipulate things without using logical operators

Print “Odd” if a given integer number is not even

Just writing `;` on its own creates an **empty statement**. No operation takes place and Mr. C takes no action on an empty statement.

Remember, **every else needs an if**, but an *if* doesn't need an *else*

<pre>#include<stdio.h> int main(){ int a = 5; if(!(a % 2 == 0)) printf("Odd"); return 0; }</pre>	<pre>#include<stdio.h> int main(){ int a = 5; if(a % 2 != 0) printf("Odd"); return 0; }</pre>	<pre>#include<stdio.h> int main(){ int a = 5; if(a % 2 == 0) ; // Empty statement else printf("Odd"); return 0; }</pre>
----------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------

Print “Leap Year” if a multiple of 4 but not multiple of 100

```
if((year%4 == 0) && (year%100 !=0)){  
    printf(“Leap Year”);  
}
```

```
if(year%4 == 0){  
    if(year%100 !=0){  
        printf(“Leap Year”);  
    }  
}
```

Set CPI to 10 if marks > 80 or if attendance >= 50 or both, else 8

```
if((marks > 80) || (att >= 50)){  
    cpi = 10;  
}else{  
    cpi = 8;  
}
```

```
cpi = 8;  
if(marks > 80){  
    cpi = 10;  
}  
if(att >= 50){  
    cpi = 10;  
}
```

Quiz: Print “Leap Year” if any one of the following happens

1. year a multiple of 4 but not multiple of 100
2. year a multiple of 400

Do the above first using logical operators and then not using them.

Some Pitfalls and recognizing compiler error messages

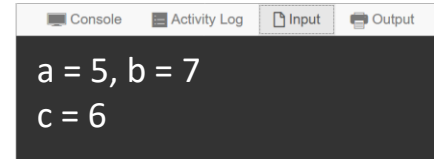
1. Mr. C. may not give compilation warnings or errors on type and operator errors (using =, &, | instead of ==, &&, || respectively).
2. When using scanf to just input numbers (int, long, float, double), can be careless about whitespaces. However, if scanf format string contains English words and other characters ('=', 'Hello', '*'), then it is better to specify every whitespace character (space, \t, \n) explicitly.

E.g if input is in two lines, then the following code will read the numbers erroneously

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

However, the following code will read things very well

```
scanf("a = %d, b = %d\nc = %d", &a, &b, &c);
```



The screenshot shows a console window with four tabs: Console, Activity Log, Input, and Output. The Console tab is active and displays the text "a = 5, b = 7" on the first line and "c = 6" on the second line, demonstrating how the second scanf call correctly reads the value of 'c' from a new line of input.

3. When using double, float variables, be prepared for some approximation errors when doing math operations like addition, multiplication, typecasting etc since Mr C does not mind making small approximations given the huge numbers he is storing. If exact digits are important, use int or long.
 4. $10 \leq x \leq 20$ may give errors and is a dangerous way to write a conditional expression. Safer way $(x \geq 10) \ \&\& \ (x \leq 20)$.
 5. Using $==$ and $!=$ comparisons with float and double variables and constants can be risky and erroneous. Use these safely with int and long. For floats, doubles, usually we check if the values are close together or not. E.g. float a, double b; Instead of saying $\text{if}(a == b)$ it is much safer to instead write $\text{if}(\text{fabs}(a - b) < 1e-5)$ where $1e-5 = 0.00001$ is some accuracy value which would be given to you.
-