

BABCOCK UNIVERSITY

ILISAN-REMO, OGUN STATE, NIGERIA.

SCHOOL OF COMPUTING AND ENGINEERING SCIENCES

DEPARTMENT OF SOFTWARE ENGINEERING

**ASSIGNMENT MARKING GUIDE**

COURSE CODE: COSC 111 COURSE TITLE: Intro to Programming in C

**SECTION A**

|  |  |
| --- | --- |
| 1 | C |
| 2 | A |
| 3 | D |
| 4 | B |
| 5 | C |
| 6 | Dennis Ritchie at Bell Laboratories in 1972 |
| 7 | There is no output |
| 8 | Run-time error |
| 9 | 6 |
| 10 | 3 |

**SECTION B**

1. Analysis of the problem identifying the output, input and processing requirements

***ANALYSIS:***

* **Output requirements:**
  + Employee Number
  + Employee Name
  + Total allowance
  + Tax
  + Pension
  + Basic salary
  + Gross pay
  + Net pay
* **Input data:**
  + Employee Number
  + Employee Name
  + Basic salary
  + Housing allowance
  + Transport allowance
* **Processing requirements:**
  + If Basic salary is <=5000, Tax = 5% of Basic salary

Else Tax = 8% of basic salary.

* + Total allowances = Housing + Transport
  + Gross Pay = Basic salary + Total allowances
  + Contributory pension= 10% of Gross Pay
  + Net pay = Gross pay – (Tax + pension)

***FLOWCHART: (System Design)***

Employee Number

Employee Name

Basic salary

Housing allowance

Transport allowance

Yes

Is basic salary <= N5000

No

Tax = 8% of basic salary

Tax = 5% of basic salary

Total allowances = Housing + Transport

Gross Pay = Basic salary + Total allowances

Pension = 10% gross pay

Net pay = Gross pay – (Tax + pension)

* + Employee Number
  + Employee Name
  + Total allowance
  + Tax
  + Pension
  + Basic salary
  + Gross pay
  + Net pay

***SYSTEM IMPLEMETATION:***

***Note: Reading in name and employee number is ignored in this program becoause the students are yet to learn the use of array for string values***

//Implemetation of monthly ***takehome*** of employees using C

/\* computing the monthly emolument of all category

of employees of the institution \*/

#include <stdio.h>

int main()

{

double basic\_sal, Housing, Transport, Meal;

double TotalAllawee, GrossPay, Tax, netPay, Pension;

printf("\nEnter basic salary of the employee: ");

scanf(“%f”, &basic\_sal);

printf("\nEnter Housing of the employee: ");

scanf("%f", &Housing);

printf("\nEnter Transport of the employee: ");

scanf("%f", &Transport);

TotalAllawee = Housing + Transport;

GrossPay = basic\_sal + TotalAllawee;

Pension = (GrossPay \* 10)/100;

if(basic\_sal <= 5000)

Tax = (basic\_sal \* 5)/100;

else

Tax = (basic\_sal \* 8)/100;

netPay = GrossPay - (Tax + Pension);

printf("\nEMPLOYEE PAY SLIP: \n");

printf("\nBasic Salary- %f", basic\_sal);

printf("\nTotal allowance- %f", TotalAllawee);

printf("\nGross pay- %f", GrossPay);

printf("\nTax- %f", Tax);

printf("\nPension- %f", Pension);

printf("\nNet Pay: %f", netPay);

return 0;

}

2) The order of evaluation of the operators in the following expression:

X = 8 + 15 \* (6 – 2) -1

R1

R2

R1

Final Result

1. What do you understand by **dangling else**

The dangling else is a problem in [programming](https://en.wikipedia.org/wiki/Computer_programming) of [parser generators](https://en.wikipedia.org/wiki/Parser_generator) in which an optional else clause in an [if–then(–else)](https://en.wikipedia.org/wiki/Conditional_(computer_programming)#if%E2%80%93then(%E2%80%93else)) statement results in nested conditionals being ambiguous.

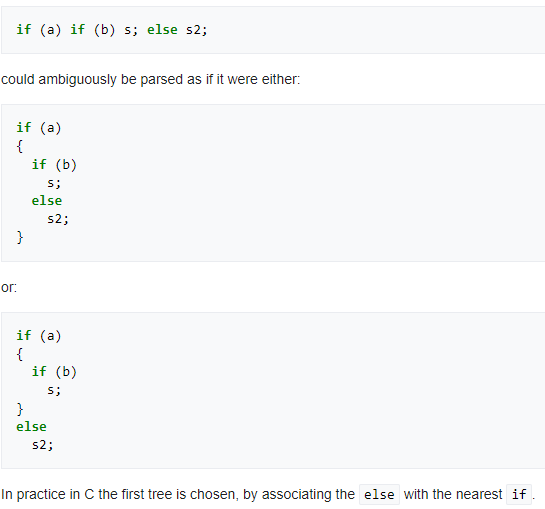
For instance: if a then if b then s else s2

This gives rise to an ambiguity in interpretation as:

if a then (if b then s) else s2

if a then (if b then s else s2)

Therefore:



1. Correctness property of an algorithm means the program must always return the desired output for all legal instances of the problem.

|  |
| --- |
| 1. #include <stdio.h> 2. int main() 3. { 4. int Grade; 5. printf("\n Enter your Grade choice of swiss lace: Grade is between 1 and 5"); 6. scanf(“%d”, &Grade); 7. switch(Grade) 8. { 9. case 1: printf("\n Grade 1 Swiss Lace: ~~#~~100, 000:00"); break; 10. case 2: printf("\n Grade 2 Swiss Lace: ~~#~~250, 000:00"); break; 11. case 3: printf("\n Grade 3 Swiss Lace: ~~#~~380, 000:00”); break; 12. case 4: printf("\n Grade 4 Swiss Lace: ~~#~~500, 000:00"); break; 13. case 5: printf("\n Grade 5 Swiss Lace: ~~#~~620, 000:00"); break; 14. default: printf("\n Option Out of Range"); 15. }//end of switch statement 16. return 0; 17. } |

6.

1. What will be the output if the first “if” evaluates as false?

**Answer**: Have a bottle of ginger-drink

1. What will be the output if the second “if” evaluates to be true?

**Answer**: Have a plate of rice

1. What will be the output if the second “if” evaluates to be false?

**Answer**: Wait for beans