

Workshop 6

In this workshop, you will code a C-language program that analyzes data logically.

LEARNING OUTCOMES

Upon successful completion of this workshop, you will have demonstrated the abilities:

- to create a simple interactive program
- to code a decision using a selection construct
- to nest a logical block within another logical block

Instructions

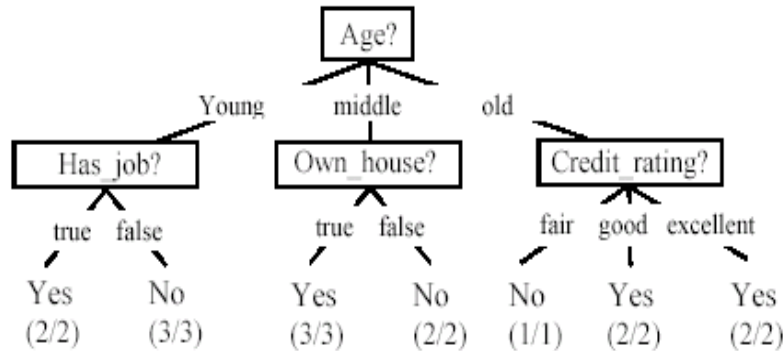
Credit card approval using decision tree classifier

Background

- A credit card company receives thousands of applications for new cards. Each application contains information about an applicant,
 - ☐ age
 - ☐ Marital status
 - ☐ annual salary
 - ☐ outstanding debts
 - ☐ credit rating
 - ☐ etc.
- Problem: To evaluate a credit card application into two categories, approved and not approved.
- Focus: implement a function to evaluate the approval (or not) of a credit card application.

Instructions

1. Input data:
The user will enter the applicant's age, has job, own_house and credit_rating. See sample data below.
2. Based on the user input, implement the decision tree shown below to decide whether to approve the credit card application or not. The program should display yes or no. Use a switch construct where most appropriate. Consider to create a mapping for categorization when using a switch construct. For instance, For Age: 0=young (ages 18-35 years), 1=mid-aged (ages 36-55), 2=old (aged older than 55 years). For Credit Rating: 0=fair, 1=good, 2=excellent



OUTPUT EXAMPLE

Test with data from lines 4, 5, 6, 8, 11, 12, and 15 in order to test the output of every single branch of the decision tree. Below is an output example for only lines 4 and 5.

ID	Age	Has_Job	Own_House	Credit_Rating	approved
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

----- Credit Card Application -----

Enter your age: 30

Do you have a job? y

Do you do you own a house? y

Enter the credit rating: Fair

approved

----- Credit Card Application -----

Enter your age: 30

Do you have a job? n

Do you do you own a house? n

Enter the credit rating: Fair

not approved

IN-LAB (30%)

In this portion you will read the data but only implement the logic for young people. Applications from middle aged or older adults will be rejected.

Submission

The in-lab portion is due by 23:59 on the day after your class. If not on matrix already, upload your `creditApproval.c` to your matrix account. Compile and run your code and make sure everything works properly.

Then run the following script from your account: (replace profname.proflastname with your professors Seneca userid)

```
~profname.proflastname/submit btm200/w6_lab <ENTER>
```

and follow the instructions. Failure to use the submitter will incur a 40% penalty.

AT-HOME (40%)

In this portion you will complete the code to make the correct decisions for the middle aged and older adults as per the table above. This code will be submitted to matrix as outlined below.

AT_HOME Reflection (30%)

Write answers to the following questions in a file called “reflect.txt” and submit it with the at-home portion of your assignment.

- How many functions did you write to complete the assignment? If you created more functions than just main, how did this benefit your program. If you put everything in main, would there be a benefit of using additional functions and, if so, what would be the benefit?
- How do you read answers that might be upper case or lower case and figure out what they user really means?

Submission

To test and demonstrate execution of your program use the same data as the output example above.

If not on matrix already, upload your `creditApproval.c` and `reflect.txt` to your matrix account. Compile and run your code and make sure everything works properly.

Then run the following script from your account: (replace profname.proflastname with your professors Seneca userid)

```
~profname.proflastname/submit btm200/w6_home <ENTER>
```

and follow the instructions.

SUBMISSION POLICY

The workshop is **due within 4 days following the in-lab assigned date by 23:59.**

All your work (all the files you create or modify) must contain your name, Seneca email and student number. Failure to use the submitter will incur a 40% penalty.

You are responsible for regularly backing up your work.

Please Note

- ☐ A successful submission does not guarantee full credit for this workshop.
- ☐ If the professor is not satisfied with your implementation, your professor may ask you to resubmit. Resubmissions will attract a penalty.