

**COURSEWORK**

Program : Diploma in Information Technology

Diploma in Computer Science

Subject : PROGRAMMING CONCEPTS & PROBLEM SOLVING

Subject code : DIT 1253

Due Date : Week 9 (21 October 2022)

|  |  |
| --- | --- |
| **NAME** | **STUDENT ID** |
| Lam Shai Yhong | 22081129 |
| Eng Zheng Yu | 22070544 |
| Chin Yung Xuan | 22057590 |
| On Duo Wei | 22092290 |
| Kydric Kok Kye Jun | 22092886 |

**Group Member Contribution Form**

**INSTRUCTION**

The contribution must be signed by all members. 3 marks will be deducted from the total marks awarded if the group fail to comply with this requirement

Group Name: Pony Co.

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** | **Student ID** | **Signature** | **Role & Responsibilities**  **(e.g. create flow chart, C++ coding etc)** |
| Lam Shai Yhong | 22081129 | Lam | C++ coding |
| Eng Zheng Yu | 22070544 | Eng | Flowchart |
| Chin Yung Xuan | 22057590 | Chin | Pseudocode, PAC, IPO |
| On Duo Wei | 22092290 | On |  |
| Kydric Kok Kye Jun | 22092886 | Kok | Presentation |

**Marking Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Exceeds Requirements** | **Meets All Requirements** | **Meets basic Requirements** | **Marks** |
|  | **13-15 marks** | **10-12 marks** | **0-9 marks** |  |
| Technical  Correctness  15% | * No technical, syntax or structure errors * Able to achieve all the anticipated result | * Some technical, syntax or structure errors * Not able to achieve some of the anticipated result | * Many technical, syntax or structure errors * Not able to achieve most or all the anticipated result |  |
|  | **6-5 marks** | **4-3 marks** | **0-2 marks** |  |
| Problem Solving Logic  6% | * Excellent problem-solving logic * Able to cater for all the scenarios. | * Some deficiency in problem solving logic or problem-solving logic is not optimum | * Many deficiencies in problem solving logic |  |
|  | **8-6 marks** | **5-3 marks** | **0-2 marks** |  |
| Solution Design (PAC & Flow chart)  8% | * Well-structured PAC * The problem well analysed and solution provided accordingly. * Flowchart developed with correct notations and right flow. * Flowchart narrates the program flow completely. | * PAC is structured. * The problem is analysed, and solution is provided. * Flowchart developed with minor errors. * Flowchart doesn’t narrate the program flow as how the program is developed. | * PAC is structured * The problem is analysed, and solution is provided. * Flowchart developed with many errors. * Flowchart doesn’t narrate the program flow as how the program is developed. |  |
|  | **3 marks** | **2 marks** | **0-1 mark** |  |
| Comments  3% | * Useful comments to elaborate the meaning of a statement / a block of statements. * Comment place in a proper manner and not overwhelming. | * Contains some comments to elaborate the meaning of a statement / a block of statements. * Comment might not be placed in a proper manner | * Contains little or no comments |  |
| Best practices  3% | * Good programming ethics and practices and implemented many good programming styles. | * Good programming ethics and practices and implemented few good programming styles. | * Poor programming ethics and practices and implemented many good programming styles. |  |
| **Comments** | | | **TOTAL (35%)** |  |

**PAC (Problem Analysis Chart)**

|  |  |
| --- | --- |
| *Given Data* | *Required results* |
| Hours parked  Parking rate  Day | Parking fee |
| *Processing Required* | *Solution Alternatives* |
| Parking fee (1st 2 hours) = Parking rate  Parking fee (3rd hour onwards) = Parking rate (for 1st 2 hours) + hours parked - 2 | 1. Define the parking rate as constants 2. Define the hours parked and parking rate as input values |

**IPO (Input Process Output)**

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| -Parking Rate  -Hours parked  -Day  -Parking Service (Normal/Valet)  -Weekday Base  -Weekend Base  -Preferred (yes/no)  -Parking ticket (yes/no) | 1. Enter parking ticket (y/n) 2. Enter parking service 3. Enter hours parked 4. Enter day 5. Enter preferred parking (y/n) 6. Calculate Parking fee  * parkingFee=(weekdayBase+hours)-2 * parkingFee=(weekendBase+hours)-2  1. Print parking fee 2. End | Parking Fee |

**Pseudocode**

START

int parkingRate, parkingFee, weekdayBase, weekendBase, total

char choice, day, preferred, parkingTicket

float hours

INPUT parkingTicket

If parkingTicket=N then

Print “Your fee is RM20”

If parkingTicket=Y then

INPUT service

If service=valet(B) then

Print “Your fee is RM15”

If service=normal(A) then

INPUT hours

If hours<=0 then

Print “Invalid option”

If hours>0 then

INPUT day

If day=Weekdays(A) OR Weekends(B) then

INPUT preferred

If preferred=Y then

weekdayBase=weekendBase=5

If preferred=N then

weekdayBase=2,

weekendBase=3

If day=Weekdays(A) then

Calculate

total=(weekdayBase+hours)-2

if total>12 then

total=12

If day=Weekends(B) then

Calculate

total=(weekendBase+hours)-2

if total>15 then

total=15

Endif

Endif

Endif

Endif

Endif

Endif

Endif

Endif

Endif

Endif

Endif

Endif

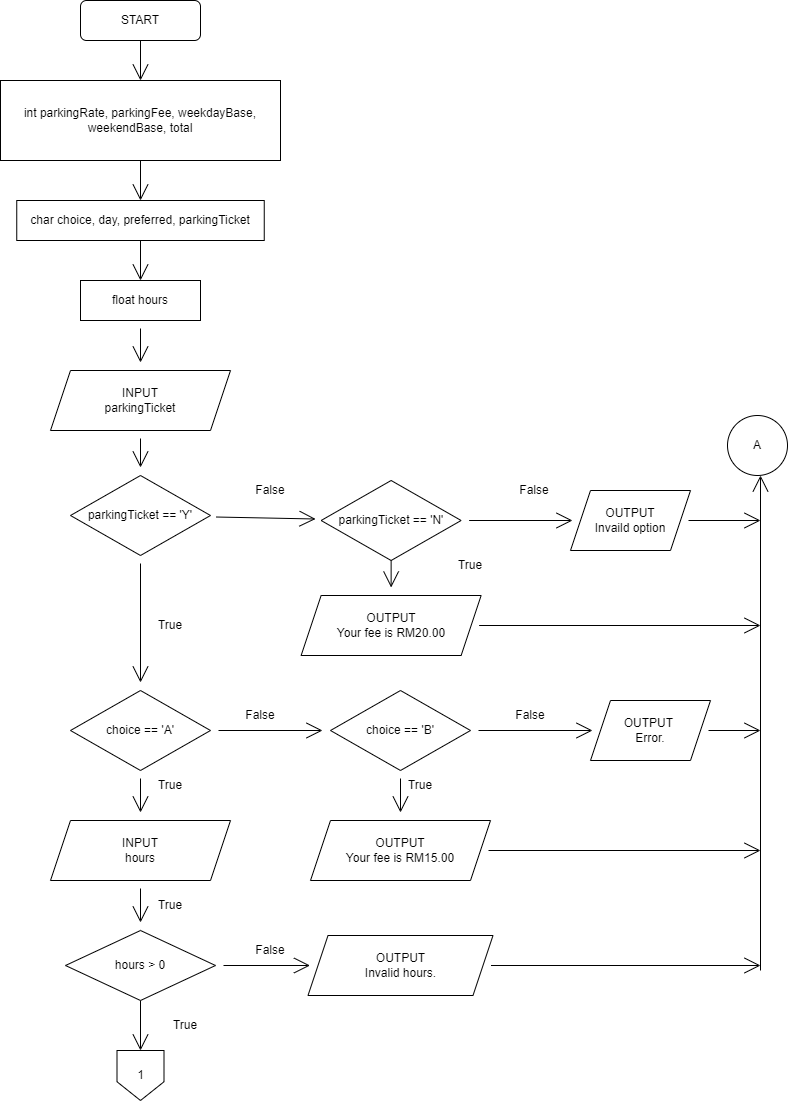
Endif

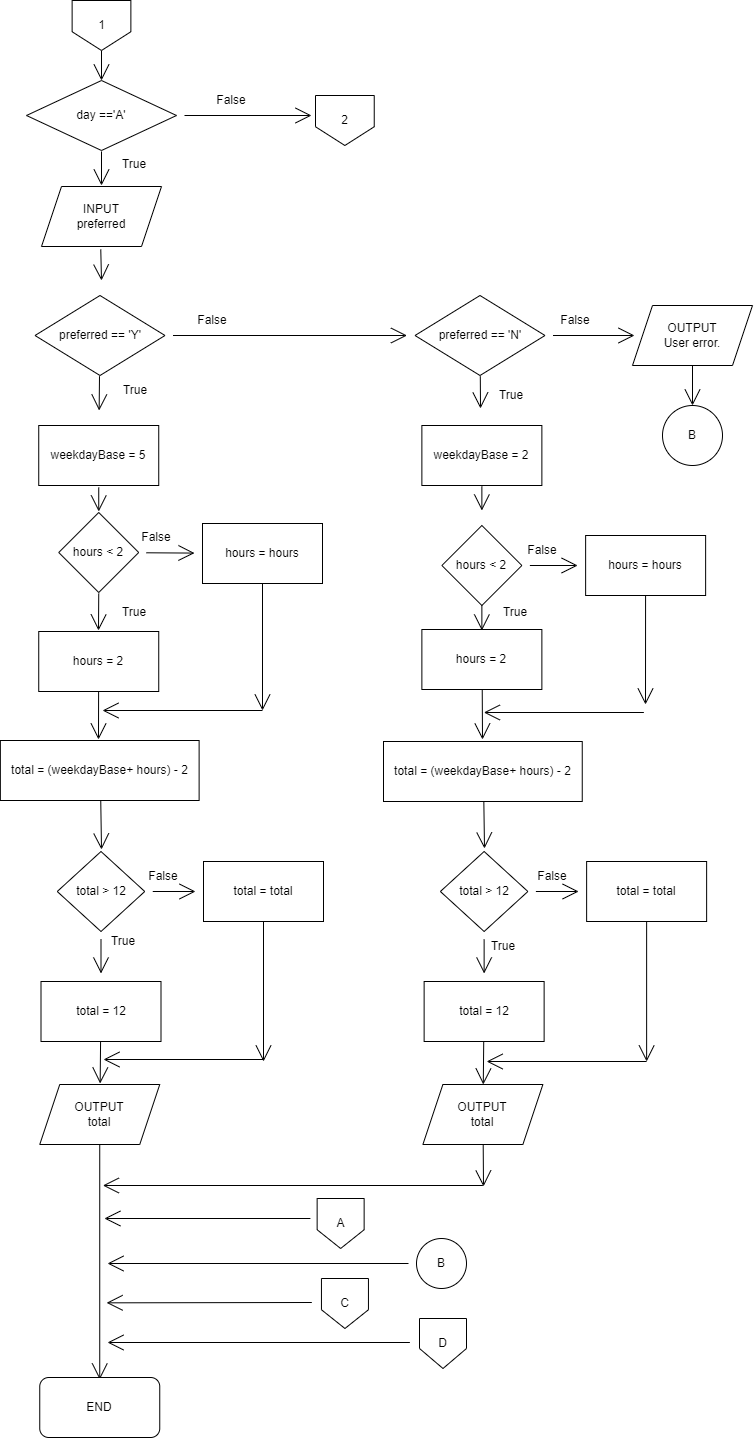
Print “Your fee is RM”

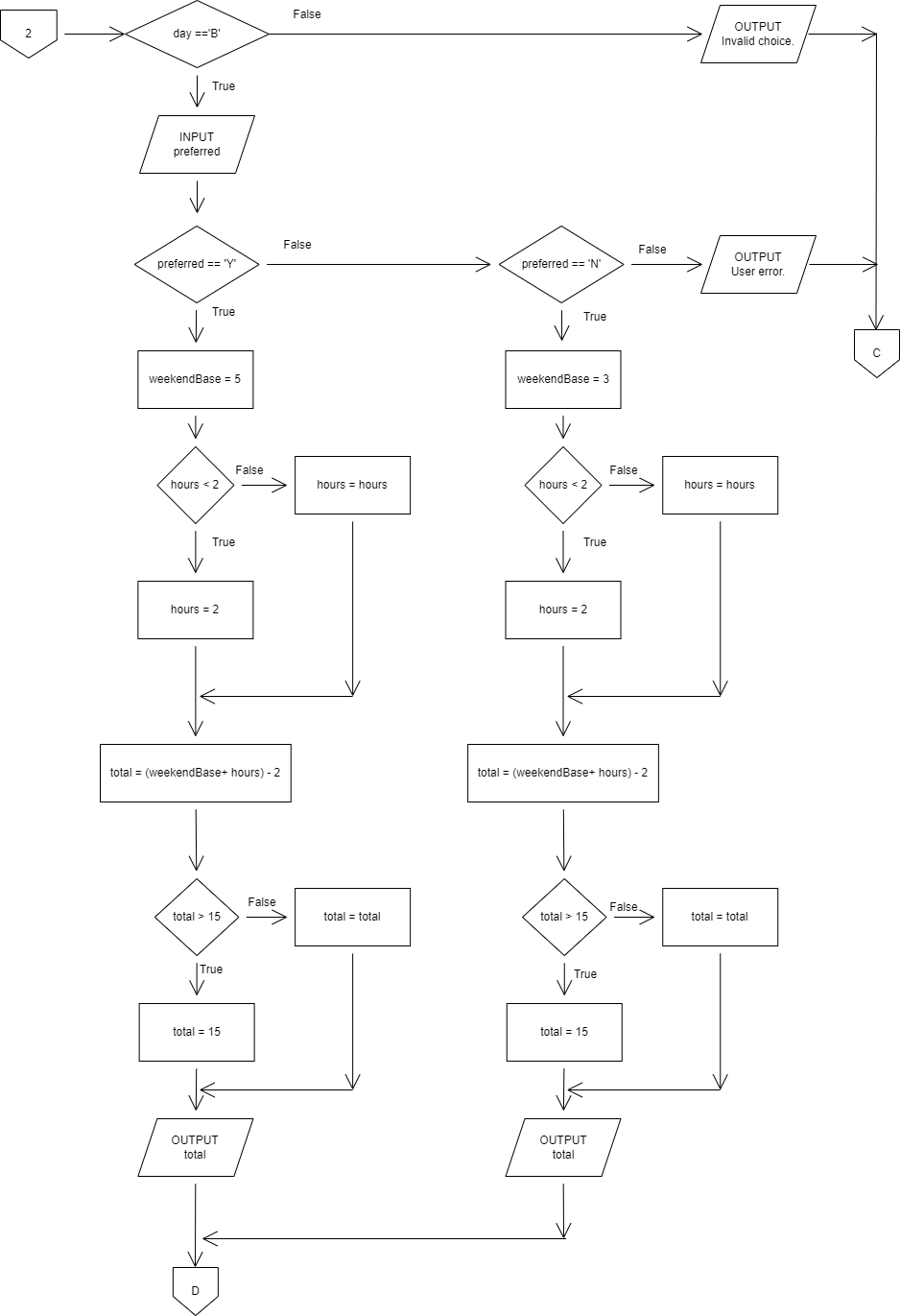
Print total

END

**Flowchart**

****

****

****

**C++ source code**

#include <iostream>

#include <cmath>

using namespace std;

//Written by Lam Shai Yhong in about 10 minutes with an extra 5 minutes of comments with about 20 minutes of actual work to fool-proof it.

int main()

{

//Defining variables.

int parkingRate, parkingFee, weekdayBase, weekendBase, total;

char choice, day, preferred, parkingTicket;

float hours;

//Lost ticket check.

cout << "Do you still own your parking ticket? Y/N\n ";

cin >> parkingTicket;

//Converting to all upper characters.

parkingTicket = (char)toupper(parkingTicket);

//Error handling.

if (parkingTicket != 'Y' && parkingTicket != 'N'){

cout << "Invalid option. (Parking)";

return 0;

}

else if (parkingTicket == 'N'){

cout << "Your fee is RM20";

return 0;

}

cout << "Choose your service.\nNormal(A)\nValet(B)\n";

cin >> choice;

//Converting to all upper characters.

choice = (char)toupper(choice);

//Switch case to decide between normal or Valet parking.

switch(choice){

case 'A':

cout << "How many hours?\n";

cin >> hours;

if (hours <= 0){

cout << "Invalid hours.";

return 0;

}

else if (hours > 0){

round(hours);

}

//Checking for weekdays or weekends.

cout << "Weekdays(A) or Weekends/Holidays(B)?\n";

cin >> day;

//Converting to all upper characters.

day = (char)toupper(day);

if (day != 'A' && day != 'B'){

cout << "Invalid choice.";

return 0;

}

//User input for preferred parking.

cout << "Preferred parking? Y/N\n";

cin >> preferred;

//Converting to all upper characters.

preferred = (char)toupper(preferred);

if (hours < 2){

hours = 2;

}

//Error handling.

if (preferred != 'Y' && preferred != 'N'){

cout << "User error.";

return 0;

}

//Preferred parking rates

else if (preferred == 'Y'){

weekdayBase = 5;

weekendBase = 5;

}

//No preferred parking rates

else if (preferred == 'N'){

weekdayBase = 2;

weekendBase = 3;

}

//Switch statement in a switch statement to determine weekday or weekends. (I was lazy.)

switch(day){

case 'A':

total = (weekdayBase + hours) - 2;

if (total > 12){

total = 12;

}

cout << "Your fee is RM" << total <<".";

break;

case 'B':

total = (weekendBase + hours) - 2;

if (total > 15){

total = 15;

}

cout << "Your fee is RM" << total <<".";

break;

default:

cout << "Invalid choice.";

}

break;

//Valet parking.

case 'B':

cout << "Your fee is RM15.00";

break;

//Error handling.

default:

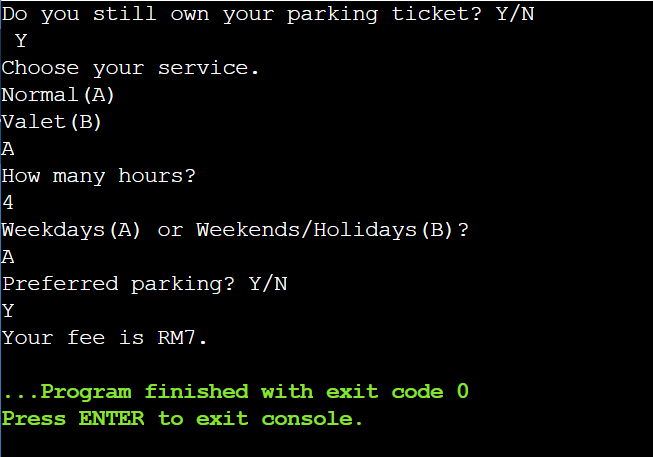
cout << "Error.";

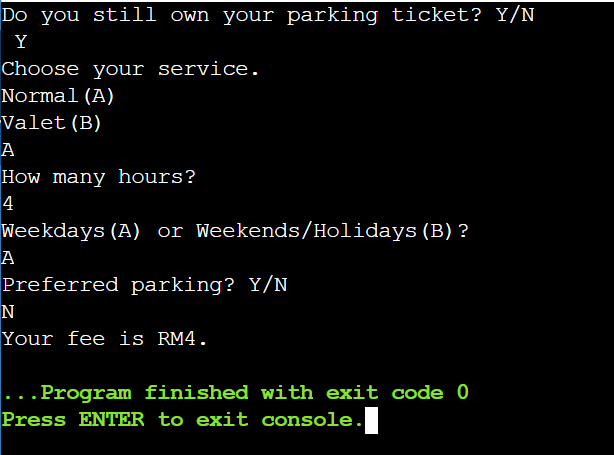
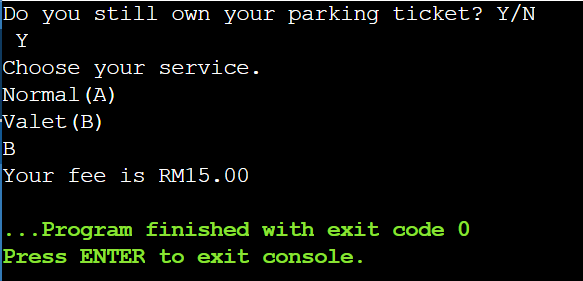
break;

}

}

**Sample output**

****

****

