5/9/2016

|  |
| --- |
| Student Number | Philip Woulfe |



|  |  |
| --- | --- |
| Soft Ireland | Procedural Programming Assignment |

Contents

[Task 1 – Report on the main advantages and disadvantages of the procedural programming paradigm 2](#_Toc450577163)

[What is the Procedural Programming paradigm? 2](#_Toc450577164)

[What are its main advantages? 2](#_Toc450577165)

[What are its main disadvantages? 2](#_Toc450577166)

[Task 2 – Report on the design specification of the payroll application 3](#_Toc450577167)

[Design specification 3](#_Toc450577168)

[Diagrams 3](#_Toc450577169)

[Inputs 3](#_Toc450577170)

[Outputs 3](#_Toc450577171)

[Any other relavent information 3](#_Toc450577172)

[Task 3 – Develop application 4](#_Toc450577173)

[Task 4 – Report on testing 5](#_Toc450577174)

[Test of the application 5](#_Toc450577175)

[Recommendation for improvements 5](#_Toc450577176)

[Support document 5](#_Toc450577177)

# Task 1 – Report on the main advantages and disadvantages of the procedural programming paradigm

## What is the Procedural Programming paradigm?

Procedural programming (PP) is a programming paradigm derived from the structural programming paradigm. The core concept of procedural programming is to break down the program into procedures (comparable with Object Oriented (OOP) methods) to make the program more modular. Modularity is desirable as it allows a team of developers to collaborate more easily on a large project. The modularity allows different developers to work on separate self-contained procedures without disrupting the rest of the program.

### Features shared with other programming paradigms

#### Variables

Variables are used to store information. These can be primitive variables such as:

* natural numbers
* characters
* floating point numbers
* booleans

#### Structs

Structs are custom data-types, and example of this would be a *string*, under the surface this is just a char array.

#### Procedures

Procedures are a way of manipulating data. This can be procedures that take input and providing output or it can be structures like *if* statements or *loops*. These procedures are the key feature of procedural programing. Before procedures, goto statements were used to reuse other parts of code. This made code to read and understand what it was doing, as it could access any other part of the program. It could also have unexpected consequences as lines of code you didn’t want rerun could be executed.

### Differences from OOP

Where OOP bundles both procedures and data into objects, PP considers procedures and data as two separate concepts. Libraries are used to define data-types (structs) and procedures which can then be reused as needed. This is similar to OOP’s use of objects, except the data types only ever contain information, not programming logic.

#### Modularity

Procedures (Methods)

Inputs – arguments

Outputs – return values

#### Scoping

## What are its main advantages?

* Ubiquity – they are so prevelant that they are the language with the most resources available for learning.
* They are good for general purpose programming.
* They are more portable than low level languages which have to be written for specific CPUs

## What are its main disadvantages?

* They are less efficient than custom build low-level programs. This is only really an issue in a system where there are limited resources.
* Since there are so many different languages, programmers generally have to specialise in a particular language.

# Task 2 – Report on the design specification of the payroll application

## Design specification

### 0001 – Employees – Add and remove

As a payroll operative, I would like to be able to add and remove employees from the application so I can keep the payroll files up to date with current employees.

#### Happy Case:

|  |  |
| --- | --- |
| Criteria 1 | Adding an employee allows users to calculate payroll for that employee. |
| Criteria 2 | Removing an employee prevents users from calculating payroll for that employee. |

#### Error Validation:

|  |  |
| --- | --- |
| Criteria 1 | Employee names should be unique. |
| Criteria 2 | If an employee is removed, you should no longer be able to calculate payroll for that employee. |
| Criteria 3 | If an employee is removed, they should no longer appear on the payroll output file . |

### 0002 – Employees – Department

As a payroll operative, I would like to be able to assign each employee a department so I can know which department each employee works in.

#### Happy Case:

|  |  |
| --- | --- |
| Criteria 1 | Each employee must be assigned a department on creation. |
| Criteria 2 | The department should be updateable in case the employees move departments. |
| Criteria 3 | Department must be either HR, IT, Production or Sales & Marketing. |

#### Error Validation:

|  |  |
| --- | --- |
| Criteria 1 | Selecting a department outside the above should display an error. |

### 0003 – Employees – Rate of pay

As a payroll operative, I would like to be able to assign each employee a rate of pay so I can calculate their wages.

#### Happy Case:

|  |  |
| --- | --- |
| Criteria 1 | There should be three bands of pay. |
| Criteria 2 | An employee can change the rate of pay between bands. |
| Criteria 3 | Department must be either HR, IT, Production or Sales & Marketing. |

#### Error Validation:

|  |  |
| --- | --- |
| Criteria 1 | Selecting an incorrect level of pay should display an error. |

### 0004 – Employees – Wages

As a payroll operative, I would like to be able to calculate wages for each employee so I can issue payments weekly.

#### Happy Case:

|  |  |
| --- | --- |
| Criteria 1 | I should be able to enter weekly hours worked for each current employee |
| Criteria 2 | Gross wages should be calculated as hours worked \* hourly rate |
| Criteria 3 | The smallest allowable unit of work should be no less than ¼ of an hour |

#### Error Validation:

|  |  |
| --- | --- |
| Criteria 1 | I should not be able to enter hours for non-employees |

### 0005 – Employees – I/O

As a payroll operative, I would like to be able to issue a weekly statement of wages for the purposes of record keeping.

#### Happy Case:

|  |  |
| --- | --- |
| Criteria 1 | Application should read employee list from a file. |
| Criteria 2 | Application should save updated employee list to a file. |
| Criteria 3 | Once the wages are calculated, the result should be output to a file. |
| Criteria 4 | The name of the output file should include the date of the current week. |
| Criteria 5 | If more than one file is produced in a week, the newer file should overwrite the older file. |

#### Error Validation:

|  |  |
| --- | --- |
| Criteria 1 | Should generate error if can’t open employee list |
| Criteria 2 | Should create employee list if can’t find employee list |
| Criteria 3 |  |
|  |  |

## Diagrams

## Custom Data Types

Struct {

String Name

Enum Dept

Enum Rate

Boolean CurrentEmployee

} Employee

## Inputs

NewEmployee(name, dept, rate)

ChangeDept(name, dept)

ChangeRate(name, rate)

HireOldEmployee(name)

TerminateEmployee(name)

## Outputs

SaveEmployeesToFile(Employee[])

ProcessWages(Employee[])

## Any other relevant information

# Task 3 – Develop application

# Task 4 – Report on testing

## Test of the application

## Analysis of actual test results against expected results to identify discrepancies

### 0001 – Employees – Add and remove

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Customer Password Validation (Sprint 2)** | | | | | |
| **Test Case ID** | **Condition** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Comments** |
| 1.1 | Entering valid matching customer username and password from database and click Login | Allows access to the customer section of the application | As Expected | Pass |  |
| 1.2 | Clicking cancel | Exits Application | As Expected | Pass |  |
| 1.3 | Enter incorrect username or password and click Login | Display error message: “Invalid username or password” | As Expected | Pass |  |
| 1.4 | Leave username field blank and click Login | Display error message: “User name cannot be blank” | As Expected | Pass |  |
| 1.5 | Leave password field blank and click Login | Display error message: “Password cannot be blank” | As Expected | Pass |  |
| 1.6 | If the database cannot be reached | Display error message: "Unable to connect to Database. Please ensure database exists and there are no currently open connections" | As Expected | Pass |  |
| 1.7 | Help | Clicking help opens the instruction manual | As Expected | Pass |  |

### 0002 – Employees – Department

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Administrator Password Validation** | | | | | |
| **Test Case ID** | **Condition** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Comments** |
| 2.1 | Entering valid matching administrator username and password from database and click Login | Allows access to the maintenance section of the application | As Expected | Pass |  |
|  |  |  |  |  |  |

### 0003 – Employees – Rate of pay

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Booking Flights** | | | | | |
| **Test Case ID** | **Condition** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Comments** |
| 3.1 | Navigate to customer screen | Departure airport, destination airport should be selectable for each leg of the journey | As Expected | Pass |  |
| 3.2 | Navigate to customer screen | Seat type should be selectable for the flights | As Expected | Pass |  |
| 3.3 | Navigate to customer screen | Departure and destination airports should be displayed in alphabetical order. | As Expected | Pass |  |
| 3.4 | Navigate to customer screen | It should be possible to book a second leg, onward from the first destination or back to the original departure airport. | As Expected | Pass |  |
| 3.5 | Navigate to customer screen | There should be three choices of seat: Economy, Business & First Class | As Expected | Pass |  |
| 3.6 | Navigate to customer screen | Departure and destination airport cannot be the same. | As Expected | Pass |  |
| 3.7 | Navigate to customer screen, leave an enabled combo blank and click Book Flights | Error Message: “Please select all airport and seat type” | As Expected | Pass |  |
| 3.8 | Navigate to the customer screen, click clear | All combo boxes, check boxes and radio buttons should return to their default state. | As Expected | Pass |  |
| 3.9.1 | Navigate to customer screen, click Logout. | User should be returned to the Login Screen | As Expected | Pass |  |
| 3.9.2 |  | Password Field should be blank after logout | As Expected | Pass |  |
| 3.10 | Help | Clicking help opens the instruction manual | As Expected | Pass |  |

### 0004 – Employees – Wages

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Confirming Flights** | | | | | |
| **Test Case ID** | **Condition** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Comments** |
| 4.1 | Select flights and seats, click Book Flight | The passenger name and booking reference should be recorded on the Confirmation Screen | As Expected | Pass |  |
| 4.2 | Select flights and seats, click Book Flight | Details of flight should be displayed for the user to confirm | As Expected | Pass |  |
| 4.3 | Navigate to confirm screen, click confirm | Details of the flight should be written to the database | As Expected | Pass |  |
| 4.4 | Navigate to confirm screen, click confirm | Clicking confirm disables the confirm button and enables print button | As Expected | Pass |  |
| 4.5 | Navigate to confirm screen, click print | Booking details should be printed to the default printer if one exists | As Expected | Pass |  |
| 4.5 | If the information cannot be written to the database | Display error message | As Expected | Pass |  |
| 4.6 | Help | Clicking help opens the instruction manual | As Expected | Pass |  |
| 4.7 | Exit confirm screen and re-enter | Confirm enabled, print disabled | As Expected | Pass |  |
|  |  |  |  |  |  |

### 0004 – Employees – I/O

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Maintenance Suite** | | | | | |
| **Test Case ID** | **Condition** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Comments** |
| 5.1 | Login to application with admin password, select airport from dropdown list, click remove. | Airport should be removed from this combo box, the combo boxes in the customer screen and the database. | As Expected | Pass |  |
| 5.2.1 | Navigate to maintenance suite, select three letter code for new airport and fill name into text box | If the three letter code is unique, airport should appear in the application. | As Expected | Pass |  |
| 5.2.2 |  | If three letter code is not unique, display error message. | As Expected | Pass |  |
| 5.3 | Navigate to maintenance suite, flight tab | All booked flights should be viewable | As Expected | Pass |  |
| 5.4.1 | Clicking Logout | Return to Login Screen | As Expected | Pass |  |
| 5.4.2 | On return to Login Screen | Password should be blank | As Expected | Pass |  |
| 5.6 | Set three character airport code, try and create an airport with a name over 50 characters | Error Message | As Expected | Pass |  |
| 5.7 | Set three character airport code, try and create an airport without a name | Error Message | As Expected | Pass |  |
| 5.8 | Help | Clicking help opens the instruction manual | As Expected | Pass |  |

## Recommendation for improvements

## Support document