

AS1 LEVEL

FACT FILES

## Software Systems Development

For first teaching from September 2013

For first AS award in Summer 2014

For first A Level award in Summer 2015

### Introduction to Object Oriented Development Part (viii) – Testing An Object Oriented Application



software  
systems  
development

## Part (viii) – Testing An Object Oriented Application



### Learning Outcomes

#### Students should be able to:

- assess the need to apply basic testing techniques to object oriented applications;
- evaluate test plans;
- develop and use suitable test plans;
- analyse, evaluate and interpret test results.

#### The program tests should cover:

- control flow (sequential, loops);
- data flow (read / write);
- branch (method calls);
- path (IO file paths);
- statement (calculations – reasonableness of results);
- decision (if / switch);



### Content

#### Basic Testing Techniques

Basic testing techniques should be applied to the supporting classes and to the programs developed within the software system.

For each class the constructors, properties and methods should be tested to ensure that they are correctly defined. The class can then be used repeatedly with confidence in the functionality.

Common methods for programs can be tested and then re-used in subsequent programs.

For example:

```
void press_Key( int row, int col )
char againYN( int row, int col )
void message ( string message, int row, int col )
void clrMessage( int row, int col )
```

Methods written for validation of input data should be tested for all relevant criteria :

- data type - alphabetic, numeric, alphanumeric;
- format - A999 letter followed by three digits, email address;
- range - number lying between 2.00 and 15.00;
- presence - surname, forename, address exists;
- compatibility - height / weight;
- dependency - cost / selling price.

## Test Plan

A tabular format for testing is adequate at this stage.

Use a numbering system for each test section broken down for each class method.

For example: class (2)- Process Order , method (6) - Validate price

**2.6 ProcessOrder - EnterValidPrice:-** method to enter and validate a price in range 1.50 – 10.00

```
double EnterValidPrice( int row, int col )
```

No	Test Data	Reason for test	Expected outcome (manually calculated values where appropriate)	Outcome (description -only if different from expected)	Further Development
1.	10,20	Test cursor positioned to row value 10, column 20	Cursor flashes at row 10, col, 20	√	
2.	Price = xxx	Test alphabetic characters entered for price	Error message at bottom of screen, call to keyPress(), Clear message & invalid characters and cursor positioned back to row and col.	Invalid input characters not cleared	
3.	Price = 1.49	Below minimum value of range	Error message at bottom of screen, call to keyPress(), Clear message & invalid characters and cursor set back to row and col.	√	
4.	Price= 10.01	Above maximum value of range	" "	√	
5.	Price = 1.50	Minimum valid price	Accepts and positions cursor for entry of quantity	√	
....					
....					

## Evidence

Evidence such as screen dumps, reports and file contents should be cross referenced to class / method and Test Nos. e.g. 2.6.4 would reference test 4 above.

Evidence is not required for all tests carried out.

## Analysis and Evaluation

A brief description of how useful the test plan proved should be given.

A brief description should be included of areas for further development in the test plan.



## Questions

1. Develop a test plan for a method which should validate a surname of the format:-  
Range of 3 – 15 characters  
Letters only ( with exception of one embedded hyphen)
2. Develop a test plan for a method - *public static int findSmallest(int [] numbers)* which returns the smallest number in an array of integers.
3. Develop a test plan for a method - *public static double calcWage(int hrs)* which returns a wage under the following criteria:-
  - rate of pay is 6.50 for the first 35 hours worked
  - time and a half is paid for the next 8 hours and
  - double time is paid for any subsequent hours

