**Assignment # 2**

**Syed Asad Abrar**

**L164292**

**Section C**

**Question #1:**

**a. Chapter #3**

1. **a.**

Valid equivalence class:

* Any five numeric digit combination

Invalid equivalence classes:

* Any less-than five numeric digit combination
* Any greater-than five numeric digit combination
* Any length combination with non-numeric characters

Test cases:

* 54000 (Valid)
* 000 (Invalid)
* 54000 (Invalid)
* 5400a (Invalid)

**b.**

Valid equivalence class:

* 2 alphabets valid combination from the list of postal codes (<https://www.infoplease.com/us/postal-information/state-abbreviations-and-state-postal-codes>)

Invalid equivalence classes:

* More than 2 alphabets combination
* 1 alphabet
* Any length combination with numeric characters
* Any 2 alphabets combination not in the list

Test cases:

* AL (Valid)
* XY (Invalid)
* X1 (Invalid)
* ALA (Invalid)
* A (Invalid)

**c.**

Valid equivalence class:

* 1-15 alphabetic characters(can include periods, hyphens, apostrophes, spaces and numbers)

Invalid equivalence classes:

* More than 15 characters combination
* 0 length
* Any length combination with non-alphabetic characters

Test cases:

* Abrar (Valid)
* (Invalid)
* Atkinson-Lloyd (Valid)
* C00L (Invalid)
* Asdafdsffdsafdsafasdasdsad (Invalid)

**d.**

Valid equivalence class:

* Eight characters at least two of which are not alphabetic (numeric, special, nonprinting)

Invalid equivalence classes:

* Any length combination with less than two non-alphabetic characters
* Less than 8 characters
* More than 8 characters

Test cases:

* testc@s3 (Valid)
* testcase (Invalid)
* test (Valid)
* testcase123 (Invalid)

**e.**

Valid equivalence class:

* Combinations of the form, AXXXXXX where A can be from {AN, LC, RW, SM, TA, WE, WN} and X can be any digit from 0-9

Invalid equivalence classes:

* Combinations not of the form, AXXXXXX where A can be from {AN, LC, RW, SM, TA, WE, WN} and X can be any digit from 0-9

Test cases:

* AN123456 (Valid)
* ABC123 (Invalid)
* AN123 (Invalid)

**b. Chapter #4**

1. **a.**

Boundaries:

* 4 numeric digits
* 5 numeric digits
* 6 numeric digits

Test cases:

* 9999 (Invalid)
* 00000 (Valid)
* 00001 (Valid)
* 99999 (Valid)
* 000000 (Invalid)
* 000001 (Invalid)

**b.**

Boundaries:

* 0
* 50

Test cases:

* -1 (Invalid)
* 0 (Valid)
* 1 (Valid)
* 49 (Valid)
* 50 (Valid)
* 51 (Invalid)

**c.**

Boundaries:

* 1 character
* 15 characters

Test cases:

* 0 length (Invalid)
* 1 character (Valid)
* 2 characters (Valid)
* 14 characters (Valid)
* 15 characters (Valid)
* 16 characters (Invalid)

**d.**

Boundaries:

* 2 non-alphabetic characters and 6 alphabets
* 8 non-alphabetic characters and 0 alphabets

Test cases:

* 1 non-alphabetic character and 7 alphabets (Invalid)
* 2 non-alphabetic characters and 6 alphabets (Valid)
* 3 non-alphabetic characters and 5 alphabets (Valid)
* 7 non-alphabetic characters and 1 alphabet (Valid)
* 8 non-alphabetic characters and 0 alphabets (Valid)
* 9 non-alphabetic characters and 0 alphabets (Invalid)

**e.**

Boundaries:

* Valid 3 alphabet code, followed by 000000
* Valid 3 alphabet code, followed by 999999

Valid 3 alphabet codes are {PHY, EGR, ENG, LAN, CHM, MAT, PED, SOC}

Test cases:

* Valid 3 alphabet code, followed by 99999 (Invalid)
* Valid 3 alphabet code, followed by 000000 (Valid)
* Valid 3 alphabet code, followed by 000001 (Valid)
* Valid 3 alphabet code, followed by 999998 (Valid)
* Valid 3 alphabet code, followed by 999999 (Valid)
* Valid 3 alphabet code, followed by 0000000 (Invalid)

**c. Chapter # 5**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Rules** | | | | | | | | | | |
| **Conditions** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| Orphan? | Y |  |  |  |  |  |  |  |  |  |  |
| Were dependent on court until age 18? |  | Y |  |  |  |  |  |  |  |  |  |
| Are dependent on court? |  |  | Y |  |  |  |  |  |  |  |  |
| Parents alive? |  |  |  | Y |  |  |  |  |  |  |  |
| Parents married? |  |  |  | Y |  |  |  |  |  |  |  |
| Single parent? |  |  |  |  | Y |  |  |  |  |  |  |
| Parent widowed? |  |  |  |  |  | Y | Y |  |  |  |  |
| Parent remarried? |  |  |  |  |  |  | Y |  |  | Y | Y |
| Parents divorced/seperated? |  |  |  |  |  |  |  | Y | Y | Y | Y |
| Lived with one parent for more time than the other? |  |  |  |  |  |  |  | Y |  | Y |  |
| One parent provided more financial support than the other? |  |  |  |  |  |  |  |  | Y |  | Y |
| **Actions** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| Skip Step 4 | X | X | X |  |  |  |  |  |  |  |  |
| Answer questions about both parents |  |  |  | X |  |  |  |  |  |  |  |
| Answer questions about one specific parent |  |  |  |  | X | X |  | X | X |  |  |
| Answer questions about parent and stepparent |  |  |  |  |  |  | X |  |  | X | X |

**d. Chapter # 7**

We would use all transition coverage for testing enroll and drop course process.

**Enroll course:**

Test cases:

1. When class is not full

|  |  |  |  |
| --- | --- | --- | --- |
| **Current state** | **Event** | **Action** | **Next State** |
| null | create | null | Accepting Enrollment |
| Accepting Enrollment | enroll | addToSectionList; #enrolled++; | Accepting Enrollment |

1. When class has one seat left

|  |  |  |  |
| --- | --- | --- | --- |
| **Current state** | **Event** | **Action** | **Next State** |
| null | create | null | Accepting Enrollment |
| Accepting Enrollment | enroll | addToSectionList; #enrolled++; | Section Full |

1. When class is full and no waiting list exists

|  |  |  |  |
| --- | --- | --- | --- |
| **Current state** | **Event** | **Action** | **Next State** |
| Section Full | enroll | addToWaitList; #waiting=1; | Wait Listing |

1. When class is full and waiting list exists

|  |  |  |  |
| --- | --- | --- | --- |
| **Current state** | **Event** | **Action** | **Next State** |
| Wait Listing | enroll | addToWaitList; #waiting++; | Wait Listing |

**Drop course:**

Test cases:

1. When class is not full and student is enrolled

|  |  |  |  |
| --- | --- | --- | --- |
| **Current state** | **Event** | **Action** | **Next State** |
| Accepting Enrollment | drop | removeFromSectionList; #enrolled--; | Accepting Enrollment |

1. When class is not full, student is enrolled and waiting list has one student

|  |  |  |  |
| --- | --- | --- | --- |
| **Current state** | **Event** | **Action** | **Next State** |
| Accepting Enrollment | drop | remove student from SectionList; move first student from WaitList to SectionList; #waiting=0; delete WaitList; | Section Full |

1. When class is full, student is enrolled and waiting list has more than one student

|  |  |  |  |
| --- | --- | --- | --- |
| **Current state** | **Event** | **Action** | **Next State** |
| Section Full | drop | remove student from SectionList; move first student from WatList to SectionList; #waiting--; | Section Full |

1. When class is full, student is enrolled and waiting list has no student

|  |  |  |  |
| --- | --- | --- | --- |
| **Current state** | **Event** | **Action** | **Next State** |
| Section Full | drop | removeFromSectionList; #enrolled--; | Accepting Enrollment |

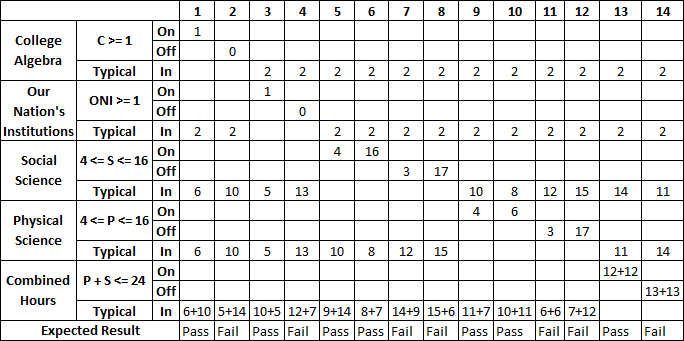
1. When waitlist has only one student who doesn’t want to be in the wait list anymore

|  |  |  |  |
| --- | --- | --- | --- |
| **Current state** | **Event** | **Action** | **Next State** |
| Wait Listing | drop | removeFromWaitList; #waiting=0; delete WaitList | Section Full |

1. When waitlist has more than one student and one of them doesn’t want to be in the wait list anymore

|  |  |  |  |
| --- | --- | --- | --- |
| **Current state** | **Event** | **Action** | **Next State** |
| Wait Listing | drop | removeFromWaitList; #waiting--; | Wait Listing |

**e. Chapter #8**



For both courses, the value 1 indicates a minimum possible number (can be anything) to pass those courses.

**f. Chapter #9**

**Test cases:**

1. **Input:** Math 1060, Mon & Wed 9:00 am, accepts (to test main success scenario)

**Process:**

A: Selects “Register for a course”

A: Selects course (Math 1060)

S**:** Displays course description

A: Selects section (Mon & Wed 9:00 am)

S: Displays section days and times

A: Accepts

S: Adds course/section to student’s course list

1. **Input:** ITUS 999 (to test first extension)

**Process:**

A: Selects “Register for a course”

A: Selects course (ITUS 999)

S**:** Displays message and exit

1. **Input:** Math 1060, Mon & Wed 9:00 pm (to test second extension)

**Process:**

A: Selects “Register for a course”

A: Selects course (Math 1060)

S**:** Displays course description

A: Selects section (Mon & Wed 9:00 pm)

S: Displays message and exit

1. **Input:** Math 1060, Mon & Wed 9:00 am, doesn’t accept (to test fourth extension)

**Process:**

A: Selects “Register for a course”

A: Selects course (Math 1060)

S**:** Displays course description

A: Selects section (Mon & Wed 9:00 am)

S: Displays section days and times

A: Doesn’t accept

S: Displays message and exit

**Question #2:**

**Question #7:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Rules** | | | | | | | | | | |
| **Conditions** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| A < B + C | F | T | T | T | T | T | T | T | T | T | T |
| B < A+ C | - | F | T | T | T | T | T | T | T | T | T |
| C < A + B | - | - | F | T | T | T | T | T | T | T | T |
| A = B | - | - | - | T | T | T | T | F | F | F | F |
| A = C | - | - | - | T | T | F | F | T | T | F | F |
| B = C | - | - | - | T | F | T | F | T | F | T | F |
| **Actions** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| Equilateral |  |  |  | X |  |  |  |  |  |  |  |
| Isosceles |  |  |  |  |  |  | X |  | X | X |  |
| Scalene |  |  |  |  |  |  |  |  |  |  | X |
| Not a triangle | X | X | X |  |  |  |  |  |  |  |  |
| Impossible |  |  |  |  | X | X |  | X |  |  |  |

**Question #3:**

**Question # 1:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Rules** | | | | | | | | | | | | | | | | | |
| **Conditions** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** |
| A < B + C | F | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T |
| B < A+ C | - | F | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T |
| C < A + B | - | - | F | T | T | T | T | T | T | T | T | T | T | T | T | T | T |
| A = B | - | - | - | T | T | T | T | F | F | F | F | F | F | F | T | F | F |
| A = C | - | - | - | T | T | F | F | T | T | F | F | F | F | F | F | T | F |
| B = C | - | - | - | T | F | T | F | T | F | T | F | F | F | F | F | F | T |
| A^2 = B^2 + C^2 | F | F | F | F | F | F | F | F | F | F | F | T | F | F | F | F | T |
| B^2 = A^2 + C^2 | F | F | F | F | F | F | F | F | F | F | F | F | T | F | F | T | F |
| C^2 = A^2 + B^2 | F | F | F | F | F | F | F | F | F | F | F | F | F | T | T | F | F |
| **Actions** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** |
| Equilateral |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Isoceles |  |  |  |  |  |  | X |  | X | X |  |  |  |  | X | X | X |
| Scalene |  |  |  |  |  |  |  |  |  |  | X | X | X | X |  |  |  |
| Not a triangle | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Right angled |  |  |  |  |  |  |  |  |  |  |  | X | X | X | X | X | X |
| Impossible |  |  |  |  | X | X |  | X |  |  |  |  |  |  |  |  |  |

**Question # 5:**

If multiple fault assumption is warranted, worst-case testing, robust worse-case testing and decision table testing are indicated. A decision table can test different conditions affecting multiple parts of a system simultaneously and gives the result consequently. With this approach, we can know if a particular set of conditions caused a fault and thus decision table can deal with multiple fault assumption.

**Question # 6:**

|  |  |  |
| --- | --- | --- |
|  | **Rules** | |
| **Conditions** | **1** | **2** |
| <= 20 mins | T | F |
| > 20 mins | F | T |
| **Actions** | **1** | **2** |
| $0.05/min | X |  |
| $1.00 + $0.01/min |  | X |

**Test cases:**

1. 19 mins

Charge: $0.95

1. 21 mins

Charge: $1.21