Roll Number: _____Section: _____

National University of Computer and Emerging Sciences, Lahore Campus

	Course:	**Software Engineering**	**Course Code:**	**CS-303**
	Program:	**BS(Computer Science)**	**Semester:**	**Fall 2017**
	Duration:	**180 Minutes**	**Total Marks:**	**80**
	Paper Date:	**14-Dec-17**	**Weight**	**40%**
	Section:	**D & E**	**Page(s):**	**9**
	Exam:	**Final**	**Reg. No.**	

Instruction/Notes:

1. State your assumptions clearly
2. Answer in the space provided
3. Answer all questions in context of class discussions, handouts and the text books.

Q1 **10+5+5= 20 Marks**

One Parking is about to launch the first version of a parking garage/lot automation project, **Reserve - Your Spot**. The parking garage currently operates without any computerized system. The management has concerns about inefficiencies of sub-optimal usage of parking space. In addition, there are frequent instances of congestion inside the garage, caused by drivers searching for vacant spots. Currently, management monitors the available locations in the garage by having employees walk around the decks to inspect if the individual spots are occupied or vacant.

The purpose of Reserve -Your -Spot is to track and manage availability of parking slots in the garage and allow registered customers to find and reserve available parking places. The reservation can be made for single day as well as multiple days. The payment of reservation shall be online or on-site through credit card only. The users can view the parking garage visually on the mobile screen. The reserved spots will be shown red however the vacant spots will be shown green. The reservation can be made either through the visual layout of the garage or user can request for a reservation and the system itself would reserve a location and notify the user through message or e-mail.

To do:

a. Specify at least 10 functional requirements of Reserve your spot system. You may specify requirements that should be in the system and are not given in the problem statement
b. Specify at least 5 non-functional requirements of Reserve your spot system. You may specify requirements that should be in the system and are not given in the problem statement

c. Since it is a new system to be built and you can easily assess from the statement if the system is large, medium, or small. Given the information about the Reserve your spot system, which process model (s) is/are the most appropriate to develop this system? Justify your answer.

Roll Number: _____ Section: _____

Roll Number: _____

@. Functional Requirements :-

FR1. System shall be able to register customers.

FR2. System shall let customer login.

FR3. System shall let customer find an available space.

FR4. System shall let customer reserve a space.

NFR1 Once reserved, system shall show the spot as red.
(look and feel: nonfunctional requirement)

FR4.1 System shall allow registeration for single day

FR.4.2 System shall allow registeration for multiple da

FR5. Once reservation period is over, system shall mark th spot as available.

NFR2. Available spots shall be shown green.

FR6. The system shall allow customers to pay for reservation

FR6.1 Payment shall be online.

FR6.2 Payment shall be onsite through credit card only.

FR7. System shall allow customers to logout.

FR8. System shall allow allow customers to see their parking

NFR3. System shall allow customers to email them th

FR9. System shall allow allow customers to reserve through

FR10. System shall allow users to register through reque

NFR4. System shall notify the user through email or

@ Nonfunctional Requirements :-

NFR5. System shall be compatible with mobiles. ? (to visua

NFR6. When registering, password length must be 8 characters. // Red write

@. Process Models :-

The two models I think would be appropriate for Reserve

Incremental Model :

1) The system is new; hence developers might not have full knowledge of domain to freeze requirements.

2) The company, as stated from in 1st paragraph, needs system due to issues hence, launch early increments.

3) Requirements are not complete hence could be incorporated later.

SCRUM

1) The earlier sprint can critical requirement and be placed on th

2) Shippable release working product.

3) System is large hence can be in sprints

4) Can focus more on

Roll Number: _____Section: _____

Q2 **8+2+5= 15Marks**

An in-car information system provides drivers with information on weather (EO), road traffic conditions (EO), and local information (EO), all on a single inquiry (EIQ). This system is linked to car radio so that information is delivered as a signal on a specific radio channel (EIF). The car is equipped with GPS receiver to discover its position and, based on that position (EI), the system accesses a couple of information services (EIFs). All information is saved in the local permanent storage (ILF)for future use and may be delivered in the driver's specified language (EI).The EIs and EIQ are considered to be of low complexity, whereas the EOs and ILF are considered to be of average complexity. One of the EIFs is simple and the other two EIFs are complex in nature. The impact of 14 characteristics (or value adjustment factors) affecting the application is given by vector F_i = [4 2 0 4 3 4 5 3 5 5 4 3 5 5].**Weighting factors are as follows:**

Component	Simple (Low complexity)	Average	Complex (High complexity)
External Inputs (EIs)	3	4	6
External Inquiries (EIQs)	4	5	7
Internal Logical Files (ILFs)	3	4	6
External Interface Files (EIFs)	7	10	15
External Outputs (EOs)	5	7	10

Function Points are calculated using the following expression:

FP = count total * [0.65 + 0.01 * sum(F_i)]

To do: Watch this video (https://www.youtube.com/watch?v=AQNhh_8fw6w)

a. Give a **Function Points (FP)** based estimate of size of the information system.

b. Using the FP based estimate of size, determine the **estimated lines of code** for the system. Assume 55 lines of code will be written per function point.

c. Based on the number of lines calculated in part b, calculate the **effort** required to complete the project where from the previous project history we know that it takes one person_month to complete 2000 lines of code (i.e. our productivity is 2 KLOC per person_month).

Roll Number: _____ Section: _____

b. Using the ~~[illegible]~~ estimate of size, determine the **estimated lines of code** for the system. Assume 55 lines of code will be written per function point.

c. Based on the number of lines calculated in part b, calculate the **effort** required to complete the project where from the previous project history we know that it takes one person_month to complete 2000 lines of code (i.e. our productivity is 2 KLOC per person_month).

(a) Function points Based Estimate:

Step 1 :- Calculate UFPs.

EOs : 3 ; Complexity = ~~4~~ 7 — $3 \times 7 = 21$

EIQs : 1 ; Complexity = 4 — $1 * 4 = 4$

EIFs : 3 ; Complexity = 7, 15 — $1 * 7 + 2 + 15 = 37$.

ILFs : 1 ; Complexity = 4 — $1 * 4 = 4$.

EIs : 2 ; Complexity = 3 — $2 * 3 = 6$

UFP = count-total = $21 + 4 + 37 + 4 + 6 = 72$.

$\sum EOs * complexity + \sum EIQs * complexity + \sum EIFs * complexity + \sum ILFs * complexity + \sum EIs * complex$

Roll Number: _____

Step 2 :- Calculate VAF

$\sum F_i$ = sum of all 14 factors

$\sum F_i = 4+2+0+4+3+4+5+3+5+5+4+3+5+5$

$\sum F_i = 52$.

Step 3 :- Put the values in the formula :-

$$FP = \underbrace{count\text{-}total}_{UFP} * \underbrace{[0.65 + 0.01 * \sum F_i]}_{VAF}$$

cm

$FP = \cancel{62}\ 72 * [0.65 + 0.01 * 52]$

$FP = \cancel{72.54}\ 84.24 \approx 85$.

(b). Estimating LOC :-

Estimating LOC from an estimated FP totally de[pends on] the "programming language."

Given 1 FP causes = 55 LOC to be written, the[n]

85 ~~84.24~~ FP = ~~85.24~~ * 55

= ~~4637.~~ 2. ≈ 4675

Almost 5KLOC ≈ (~~4633.2~~ 4675) lines of code will be written to deliver 85 FP.

(c). Estimating Effort

Productivity of a person ⇒ 2000 lines pm.

Lines of code require to deliver 85 FP ⇒ 4675.

[Tw]o [w]ays :-

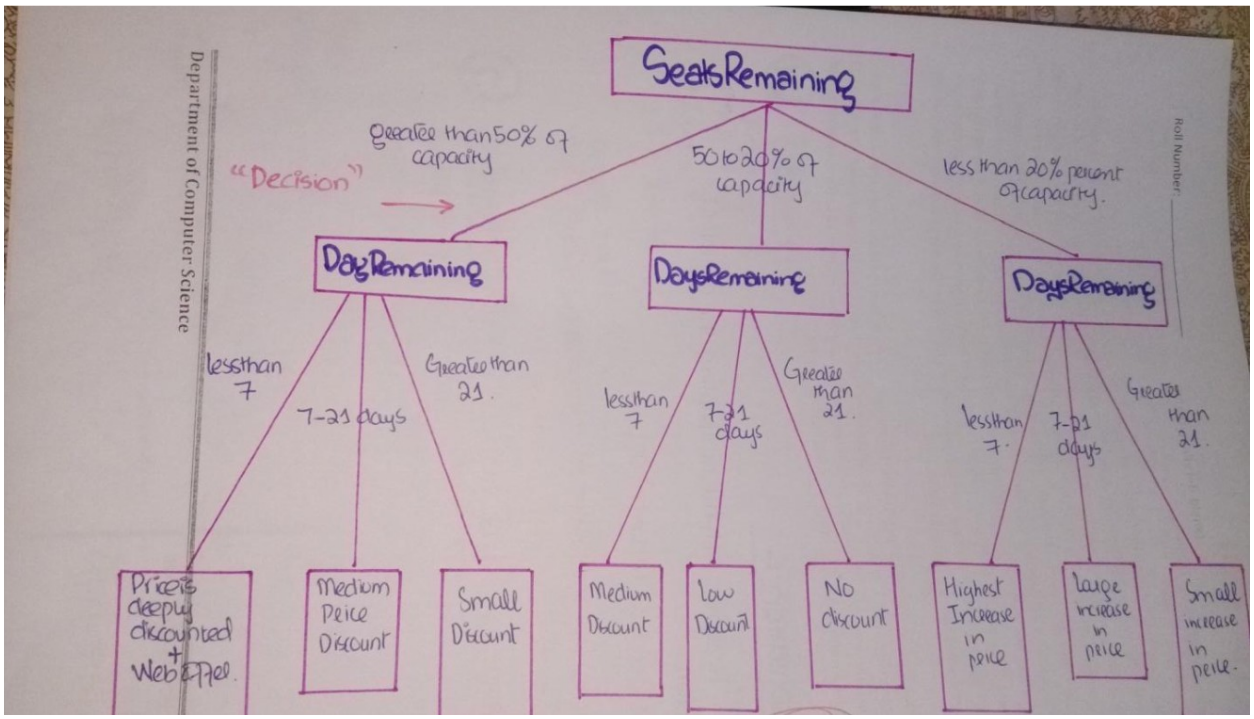
(a) Effort = $\frac{\text{lines of code}}{\text{productivity}}$

Effort = $\frac{\cancel{2000}\ 4675}{\cancel{4}\ 2000}$

(b). Effort = $\frac{}{\text{pro}}$

productivity = [illegible]

greater than 21, there is a small increase in price.



Q4 **3+1+3+3=10 Marks**

Consider the following code with line numbers mentioned:

```
1.public int binarySearch(int sortedArray[ ], int searchValue)

2.{

3. int bottom = 0, top = sortedArray.length - 1;

4. int middle, locationOfsearchValue;

5. boolean found = false;
```

Roll Number: _____Section: _____

```
int locationOfsearchValue = -1; /* the index of searchValue in the
                                   sortedArray. -1 means not found */

while ( bottom <= top && !found){
  middle = (top + bottom)/2;
  if (searchValue == sortedArray[ middle ]) {
          found = false;
          locationOfsearchValue = middle;
  }
  else
          if (searchValue < sortedArray[ middle ])
              top = middle – 1;
  }
  else
          bottom = middle + 1;
} // end while
return locationOfsearchValue;
}
```

To do:

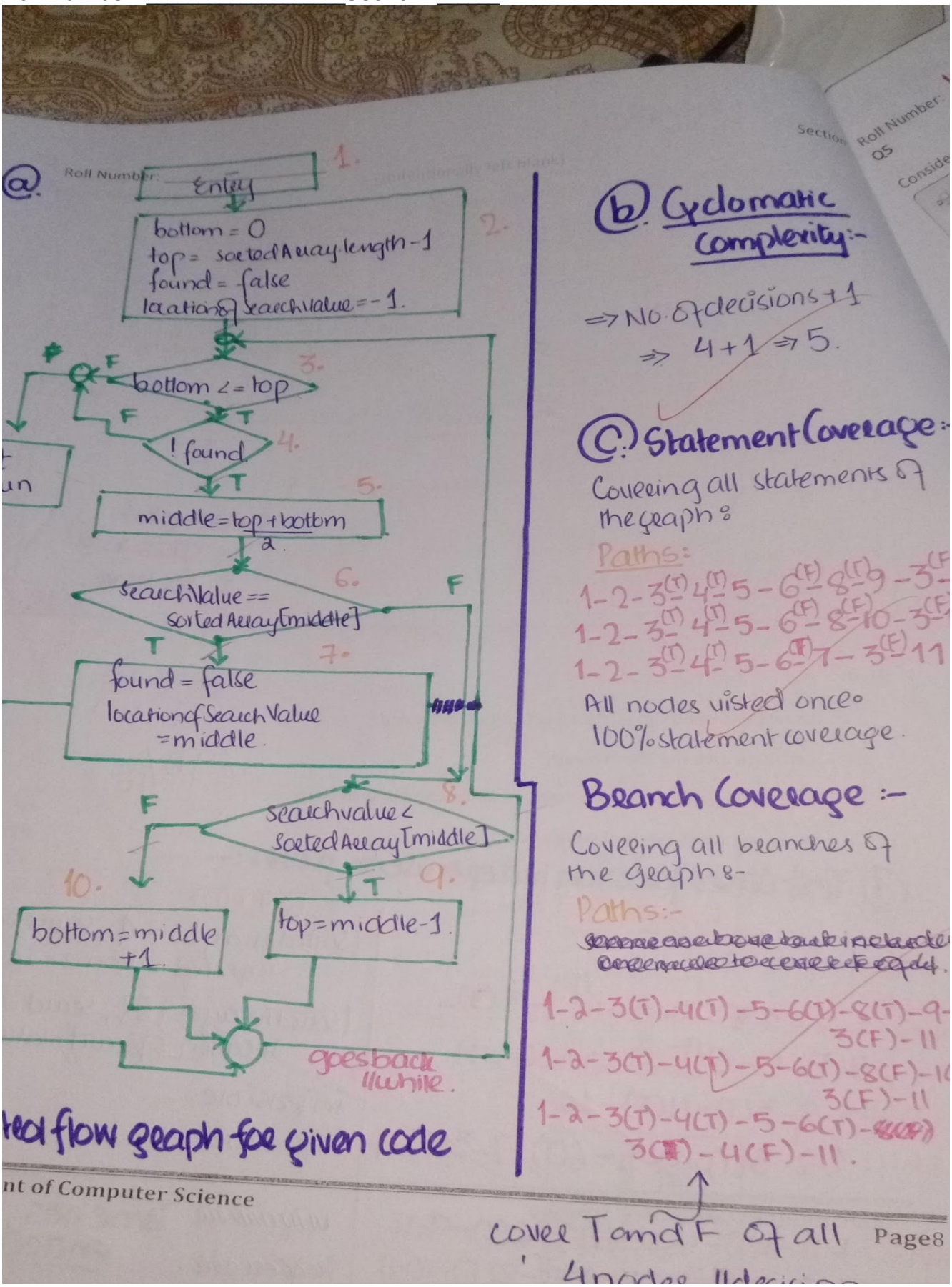
a. Draw Control flow graph or Flow Chart of above code

b. Calculate cyclomatic complexity

c. Identify all paths to achieve 100% statement and branch coverage

d. Write test cases for each independent path

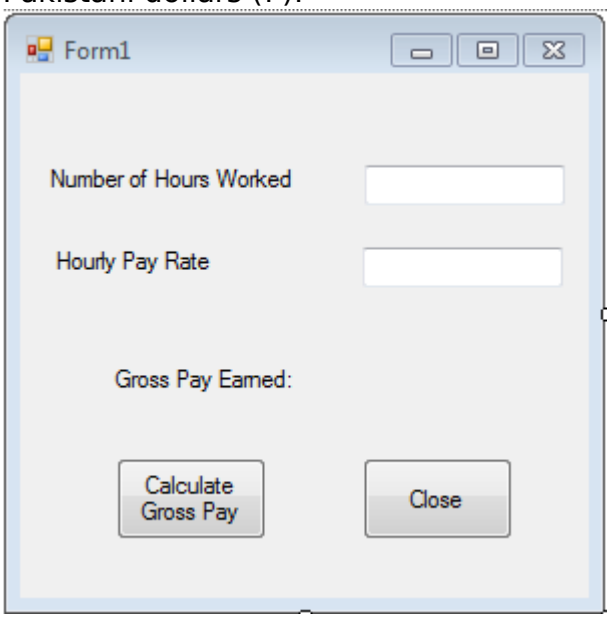
Roll Number: _____ Section: _____



Roll Number: _____Section: _____

Q5 **15 Marks**

Consider the following interface of an application that calculates weekly gross pay in Pakistani dollars (₱):



An employee cannot work for –ve numbers of hours and the maximum workload allowed is 60 hours a week. The minimum allowed hourly pay rate set by the Government is ₱ 10. The company cannot afford the pay rate above ₱ 30. The number of hours and the pay rate are rounded to the nearest integer value before their values are entered in the system. The employees are paid overtime if they work more than 40 hours a week. The pay rate remains the same but the hours above 40 are counted twice during gross pay calculations. For example if an employee has worked for 43 hours in a particular week and the pay rate for the employee is ₱ 10. Then the gross pay of the employee will be calculated as follows:

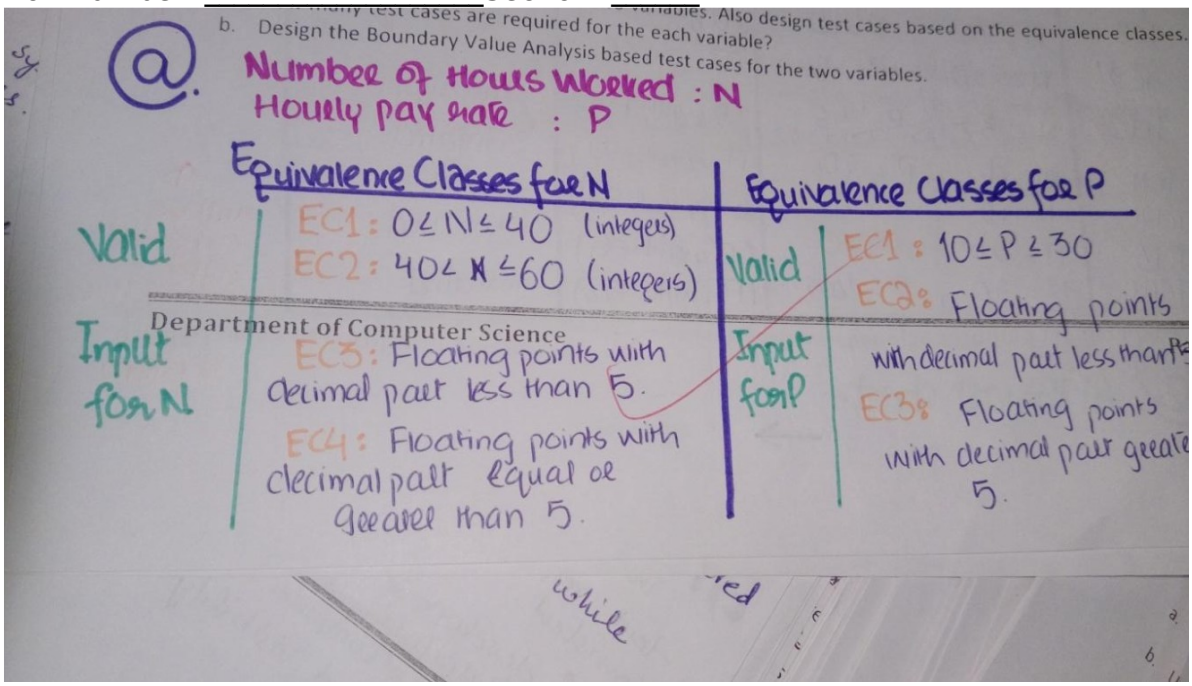
Pay for first 40 hours: 40x10 = 400 (i.e. number of hours x pay rate)
Pay for the next 3 hours: (3x2)x10 = 60 (i.e. (number of hours x 2) x pay rate)

Gross Pay Earned: 400+60 = 460

To do:

a. Design equivalence classes for the two variables. Also design test cases based on the equivalence classes. How many test cases are required for the each variable?
b. Design the Boundary Value Analysis based test cases for the two variables.

Roll Number: _____Section: _____



b. Design the Boundary Value Analysis based test cases for the two variables.

@. Number of Hours Worked : N
Hourly pay rate : P

	Equivalence Classes for N		Equivalence Classes for P
Valid Input for N	EC1: $0 \le N \le 40$ (integers) EC2: $40 < N \le 60$ (integers)	Valid Input for P	EC1: $10 \le P \le 30$ EC2: Floating points with decimal part less than 5
	EC3: Floating points with decimal part less than 5. EC4: Floating points with decimal part equal or greater than 5.		EC3: Floating points with decimal part greater 5.

Roll Number: _____Section: _____

Invalid Inputs for N

EC5: N<0

EC6: N>60

EC7: N is empty

EC8: N is an alphanumeric character

EC9: N is a string

Invalid input for P

EC5: P<10 (incorporate -ve number section)

EC6: P>30

EC7: P is empty

EC8: P is an alphanumeric character

EC9: P is a string.

One way to test the system:

1) Cross product of ECs of N with ECs of P (redundant)
2) Proceed with strong equivalent classes. Include invalid cases independently and one at a time.

Test case Identifier	Test Case values	Equivalent Classes	Expected Output	Actual Output
TC1	N=20 ; P=10	(EC_1 with EC_1)	200(GP)	
TC2	N=45 ; P=20	(EC_2 with EC_1)	1,000(GP)	
TC3	N=19.3 ; P=15 (round off to 19)	(EC_3 with EC_1)	~~300~~ 285(GP)	
TC4	N=31.5 ; P=25 (round off to 27)	(EC_4 with EC_1)	550(GP)	
TC5	N=40 ; P=10.5 (round off to 11)	(EC_2 with EC_3)	440(GP)	
TC6	N=31 ; P=14.3 (round off to 14)	(EC_2 with EC_2)	294(GP)	
TC7	N=-10 ; P=20	(EC_5 with EC_1)	invalid	
TC8	N=100 ; P=10	(EC_6 with EC_1)	invalid	
TC9	N=20 ; P=5	(EC_1 with EC_5)	invalid.	
TC10	N=35 ; P=45	(EC_2 with EC_6)	invalid	
TC11	N=0 ; P=20	(EC_7 with EC_1)	invalid	
TC12	N=10 ; P=0	(EC_1 with EC_7)	invalid	
TC13	N='ab' ; P=20	(EC_8 with EC_1)	invalid	
TC14	N=20 ; P=[illegible]			

These were demo combin i.e N=40 P=10. this is left for simi

*Cross p

Will be

Roll Number: _____ Section: _____

Invalid Inputs for N

Boundary Value Analysis

Valid ECs will be decomposed to for N:-

EC: $0 \le N \le 40$

EC1: 0 ; EC4: 39
EC2: −1 ; EC5: 40
EC3: 1 ; EC6: 41.

EC: $41 \le N \le 60$

EC7: 40 (redundant)* ; EC10: 59
EC8: 41 (")* ; EC11: 60
EC9: 42 ; EC12: 61.

EC with floating points:

EC13: .4
EC14: .5
EC15: .6

Valid ECs will be decomposed for

EC: $10 \le P \le 30$

EC1: 9
EC2: 10
EC3: 11
EC4: 29
EC5: 30
EC6: 31.

∵ Some approaches take mi value of range too, but leaving it out.

Same for floating points as N.

Testcase identifier	Test values	Expected Outcomes	Actual Outcome
TC1	N = 0 ; P = 10	0	
TC2	N = −1 ; P = 11	invalid.	
TC3	N = 1 ; P = 29 (valid values)	29	
TC4	N = 39 ; P = 11	50	
TC5	N = 40 ; P = 20	800	
TC6	N = 41 ; P = 29	1160 + 58 = 1218	
TC7	N = 42 ; P = 31	invalid.	
TC8	N = 59 ; P = 9	invalid.	
TC9	N = 60 ; P = 30	1200 + 1200 = 2400	
TC10	N = 61 ; P = 10	invalid.	
TC11	N = 20.5 ; P = 10	210	
TC12	N = 19.4 ; P = 30	570	
TC13	N = 17.6 ; P = 11	220.	

only managed all values of N with some values

Roll Number: _____Section: _____

Q 6 **10 Marks**

Choose or write the appropriate answer

1. Which one is not the phase of Unified Process?
   a) Inception
   b) Elaboration
   c) Communication
   d) Construction

2. Backlog—a prioritized list of project requirements or features that provide business value for the customer. It is created when we follow the process model named
   a) Spiral Model
   b) Extreme Programming
   c) SCRUM
   d) Feature Driven Development

3. Variant of the ----------is formal system development, where a mathematical model of a system specification is created.
   a) Waterfall model
   b) Prototyping
   c) Incremental Software Development
   d) Iterative Software Development

4. Following are the types of System-testing
   a. Recovery testing
   b. Security testing
   c. Deployment testing
   d. Performance Testing

5. Non-functional requirements define system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.
6. User Stories are similar to system requirements or use cases, but focus on the user benefits, instead on system features. preferred tool in agile methods.
7. Requirements are complete when they include descriptions of all facilities required.
8. Requirements are consistent when there are no conflicts or contradictions in the descriptions of the system facilities.
9. Programmer's productivity= LOC per unit of time
10. Cyclomatic complexity (CC) remains same for a linear sequence of statements regardless of the sequence length.