# 3.2 - 2.3.3\_02: Basis Path Testing

|  |  |  |
| --- | --- | --- |
| Test Case # | Flow Graph | Code |
| 01.001 |  | if (loginAdminFunction() == 1) {  //CASE BODY  }  else {  break;  }  public static int loginUserFunction() {  System.out.println("User Name:");  strInputUserName = kybd.next();  kybd.reset();  System.out.println("Password:");  strInputPassword = kybd.next();    if (strInputUserName.equalsIgnoreCase(strUsername) && strInputPassword.equalsIgnoreCase(strPassword)) {    System.out.println("Success");  intAttemptCtr = 0; // Reset the attempt counter  return 1;  } else {  intAttemptCtr++;  System.out.println("\nUnsuccessful Attempt #" + intAttemptCtr);  if (intAttemptCtr == 3) {  System.out.println("3 Consecutive Unsuccessful Attempts.. Exiting Program");  return escape = 2;  }  return 3;  }  } |
| 01.002 |  | if (loginAdminFunction() == 1) {  //CASE BODY  }  else {  break;  }  public static int loginAdminFunction() {  System.out.println("User Name:");  strInputUserName= kybd.next();  kybd.reset();  System.out.println("Password:");  strInputPassword = kybd.next();  if (strInputUserName.equalsIgnoreCase(strAdminUsername)  && strInputPassword.equalsIgnoreCase(strAdminPassword)){  System.out.println("Success");  intAttemptCtr = 0; // Reset the attempt counter  return 1;  } else {  intAttemptCtr++;  System.out.println("\nUnsuccessful Attempt #" + intAttemptCtr);  if (intAttemptCtr == 3) {  System.out.println("3 Consecutive Unsuccessful Attempts.. Exiting Program");  return escape = 2;  }  }  return 3;  } |
| 08.001 |  | // Call to method  public static boolean advancedBooking() throws Exception {  // Add a customer  // Make a booking  // Add a showing to the booking  if (customer.getIntAge() > newBooking.getFilmShowing().getIntRating()) {  System.out.println("Customer Under Age");  return false;  } else {  // Add booking and choose seats return true;  } |
| 03.002 |  | // CREATE CUSTOMER OBJECT  customers.addCustomer(customer);  Booking newBooking = new Booking(customer, stf1);  System.out.println("Available Showings");  showings.showShowings();  System.out.println("ID Of Showing");  kybd.reset();  intCase = kybd.nextInt();  newBooking.setIntShowingID(showings.getByID(intCase));  newBooking.setIntFilmRating(showings.getByRating(intCase));  newBooking.setDblTotalPrice(showings.getByPrice(intCase));  if (customer.getIntAge() < newBooking.getIntFilmRating()) {  System.out.println("Customer Under Age");  return false;  } else {  Seats seatBooking = new Seats(newBooking);  System.out.println("Number of Seats: ");  intCase = kybd.nextInt();  newBooking.setDblTotalPrice(newBooking.getDblTotalPrice() \* intCase);  if (intCase > 10 || intCase <= 0) {  System.out.println("Exceeds maximum number of seats per booking");  return false;  }  if (intCase <= 0) {  System.out.println("Number of seats must be greater than 0");  return false;  }  seatBooking.setIntNumSeats(intCase);  System.out.println("Row: (A - I)");  strCase = lineReader.readLine();  switch (strCase.toUpperCase().charAt(0)) {  case 'A':  seatBooking.setObjRow(Row.A);  break;  // ETC  case 'I':  seatBooking.setObjRow(Row.I);  newBooking.setDblTotalPrice(newBooking.getDblTotalPrice() \* 1.5);  break;  default:  System.out.println("Invalid Seat");  return false;  }  System.out.println("Column: (0 - 15)");  intCase = kybd.nextInt();  switch (intCase) {  case 0:  seatBooking.setObjColumn(Column.J);  break;  // ETC  default:  System.out.println("Invalid Seat");  return false;  }  bookings.addBooking(newBooking);  screen1.addSeat(seatBooking);  System.out.println(newBooking);  System.out.println(screen1);  return true;  }  } |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |