**Java classes and methods**

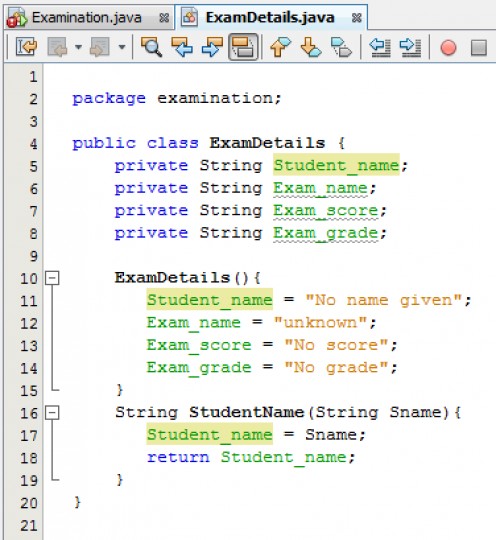
In the last lesson we learnt how to create **Java constructors** and **setter methods**. We saw how we can use constructors to set default values and how we can use setter methods to assign new values that overwrite the default values.

In this lesson, we shall continue from where we left of and learn more on what we can do with Java classes. Do NOT try this lesson unless you have completed the previous lessons on CLASSES and Constructors.

We are going to add three more methods in the **Examination** class. Each of these methods will perform a specific function.

One method will return the **Examination name** another will return the **score** while the last method will return the **grade**.

We shall also use control structures such as **IF ... ELSE statements**to add functionality to these methods. Remember the final code we had in the class **ExamDetails**? Here, have a look at it again:



Still in the **ExamDetails**class we are going to add another method that will return the **Examination name**. The method will receive a two characters input and use **IF … ELSE statement** to return the full examination name. For example if we pass “**VB**” the method should return “**Visual Basic .NET**”. Add the following code in the **ExamDetails** class. You can either type or copy and paste the code.

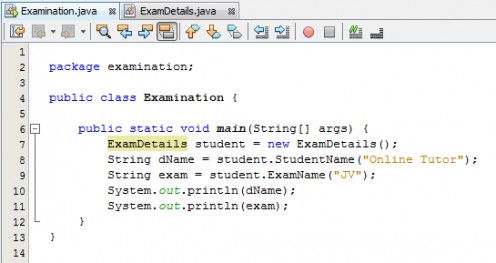
**Java classes and methods**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | package examination;    public class ExamDetails {      private String Student\_name;      private String Exam\_name;      private String Exam\_score;      private String Exam\_grade;        ExamDetails(){         Student\_name = "No name given";         Exam\_name = "unknown";         Exam\_score = "No score";         Exam\_grade = "No grade";      }      String StudentName(String Sname){         Student\_name = Sname;         return Student\_name;      }      String ExamName(String Ecode){          //Return examination name according to code passed          if(Ecode.equals("VB")){              Exam\_name = "Visual Basic .NET";            }else if(Ecode.equals("JV")){              Exam\_name = "Java .NET";            }else if(Ecode.equals("PH")){              Exam\_name = "PHP";            }else{              Exam\_name = "No Exam selected"; //Incase no code was entered          }          return Exam\_name;      }  } |

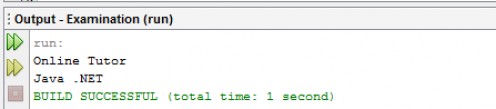
In the above code, the method **ExamName** receives a two characters input representing the examination name.

Using **IF … ELSE statement**, it returns the full examination name representing the two characters input else it returns the string "**No Exam selected**".

In the **Examination** class i.e. the main class, add the code to pass the Examination two characters input (**Ecode**) and the output statement like shown below:



At this stage, run the main class again and this time you expect it to display the **Student name** and **Examination name**.

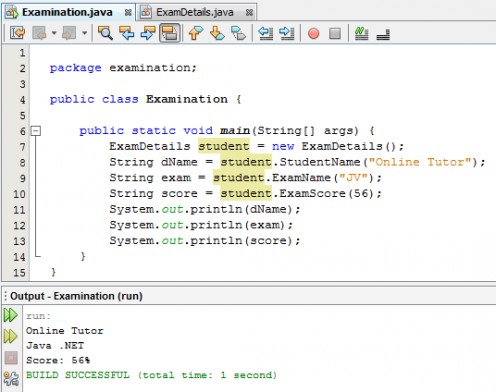


Just like what we have done with the **ExamName** method, we are going to add another method that will return the examination score. Add the following code in the **ExamDetails** class.

**Java classes and methods**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | package examination;    public class ExamDetails {      private String Student\_name;      private String Exam\_name;      private String Exam\_score;      private String Exam\_grade;        ExamDetails(){         Student\_name = "No name given";         Exam\_name = "unknown";         Exam\_score = "No score";         Exam\_grade = "No grade";      }      String StudentName(String Sname){         Student\_name = Sname;         return Student\_name;      }      String ExamName(String Ecode){          //Return examination name according to code passed          if(Ecode.equals("VB")){              Exam\_name = "Visual Basic .NET";            }else if(Ecode.equals("JV")){              Exam\_name = "Java .NET";            }else if(Ecode.equals("PH")){              Exam\_name = "PHP";            }else{              Exam\_name = "No Exam selected"; //Incase no code was entered          }          return Exam\_name;      }      String ExamScore(int Escore){          Exam\_score = "Score: "+Escore + "%"; //Get score          return Exam\_score;      }  } |

The **ExamScore** method combines the **score** with strings “**Score:**“ and “**%**”. So if the value in **Escore** is **67**, the text "**Score: 67%"** will be stored in the **Exam\_score** field. In the **Examination** class i.e. the main class, add the code to pass the score (**Escore**) and the output statement like shown below. Run the program again to see the output.

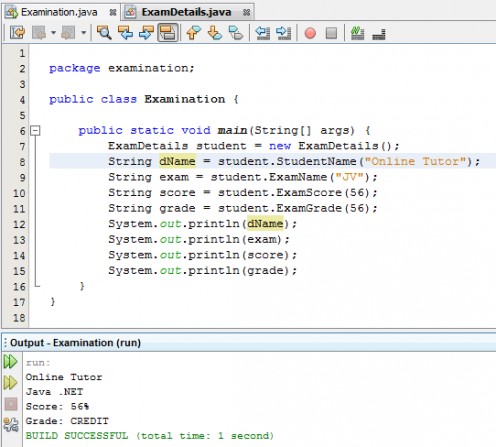


Now for the final part, we are going to add another method that will return the examination grade according to the score attained. Add the following code in the **ExamDetails** class.

**Java classes and methods**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54 | package examination;    public class ExamDetails {      private String Student\_name;      private String Exam\_name;      private String Exam\_score;      private String Exam\_grade;        ExamDetails(){         Student\_name = "No name given";         Exam\_name = "unknown";         Exam\_score = "No score";         Exam\_grade = "No grade";      }      String StudentName(String Sname){         Student\_name = Sname;         return Student\_name;      }      String ExamName(String Ecode){          //Return examination name according to code passed          if(Ecode.equals("VB")){              Exam\_name = "Visual Basic .NET";            }else if(Ecode.equals("JV")){              Exam\_name = "Java .NET";            }else if(Ecode.equals("PH")){              Exam\_name = "PHP";            }else{              Exam\_name = "No Exam selected"; //Incase no code was entered          }          return Exam\_name;      }      String ExamScore(int Escore){          Exam\_score = "Score: "+Escore + "%"; //Get score          return Exam\_score;      }        //Here we find grade      String ExamGrade(int Escore){          String grade = "";          if(Escore >=0 && Escore <= 35){              grade = "FAIL";            }else  if(Escore >=35 && Escore <= 50){              grade = "PASS";            }else  if(Escore >=50 && Escore <= 70){              grade = "CREDIT";            }else  if(Escore >=70 && Escore <= 100){              grade = "DISTINCTION";            }else{              grade = "Score out of range";            }          return "Grade: "+grade;      }  } |

In the main class, add the code to pass the score and output the grade just like we did with the other methods. The complete main class i.e. **Examination** class should look like shown below. Run the main class and you should be able to get the Student name, Examination name, Score and Grade.



**Sample Java problem with copy and paste solution code**

Write a Java application which meets the following requirements:

In the application, you should declare a class called “**Employee**” with the following details:

**Variables**

| **Variable Name** | **Description of variable** |
| --- | --- |
| EmpNo | Employee Number |
| EName | Employee Name |
| EDesig | Employee Designation |
| BSal | Basic Salary |
| HA | House Allowance |

**Methods**

| **Member Functions** | **Description of Member Functions** |
| --- | --- |
| getValues() | Should initialize the values for the member variables EmpNo, ENname, EDesig, BSal, HA |
| CalculateSalary() | Should calculate the Gross salary as the sum of the BSal and HA |
| DisplayValues() | Should print the value of the instance variables along with Gross salary |

**Requirements**

1. In your main method, create **N** number of Employee objects, where the value of N is obtained from the user and store the objects into an array.
2. Use appropriate methods to read the values of the Employee objects from user, Calculate the Gross Salary for each Employee and print employee details and salary details of all objects created.
3. Use exception handling appropriately.
4. Use comments to illustrate the various concepts applied / utilized in the solution.
5. Ensure the use of meaningful variable names, consistent indentation of program code

**Java solution code for the above problem**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58 | /\*   \* This program accepts salary details   \* of N number of employees, calculates gross salary   \* and print the final details on the console   \*/  package employee;  import javax.swing.JOptionPane;  public class Employee {      private int EmpNo; //Employee Number      private String EName; //Employee name      private String EDesig; //Employee designation      private int BSal; //Basic salay      private int HA; //House allowance      private int GSal; //Gross salary        Employee() { //Create Employee class constructor and set default values          EmpNo = 0;          EName = "Unknown";          EDesig = "Unknown";          BSal = 0;          HA = 0;          GSal = 0;  }       void getValues(){ //Initialize instance variables          EmpNo = Integer.parseInt(JOptionPane.showInputDialog("Enter employee number"));          EName = JOptionPane.showInputDialog("Enter employee name");          EDesig = JOptionPane.showInputDialog("Enter designation");          BSal = Integer.parseInt(JOptionPane.showInputDialog("Enter basic salary"));          HA = Integer.parseInt(JOptionPane.showInputDialog("Enter House allowance"));      }     int CalculateSalary(){ //Calculate basic salary         GSal = BSal+HA;         return GSal;     }     void DisplayValues(){ //Print values of the instance variables           System.out.println("Employee nunber: "+EmpNo);         System.out.println("Employee name: "+EName);         System.out.println("Designation: "+EDesig);         System.out.println("Basic salary: "+BSal);         System.out.println("House allowance: "+HA);         System.out.println("Gross salary: "+this.CalculateSalary());     }      public static void main(String[] args) {        //Get N number of employees from the user        int EmpNum = Integer.parseInt(JOptionPane.showInputDialog("Calculate salary for how many employees?"));        Employee[] newEmp = new Employee[EmpNum];//Create array of N objects          try { //Trap and handle errors if any              for(int i = 0; i<newEmp.length; i++){                   newEmp[i] = new Employee(); //Loop through the objects and for                   newEmp[i].getValues(); //every object, get employee details        }        System.out.println("----------------------------------------------");        System.out.println("EMPLOYEES SALARY DETAILS. TOTAL EMPLOYEES: "+EmpNum);                for(int i = 0; i<newEmp.length; i++){                  System.out.println("----------------------------------------------");                  newEmp[i].DisplayValues(); //Loop through the objects and print details              }        }catch (Exception err ) {              System.out.println(err.getMessage());//Throw error message if any          }       }    } |

**Here is a Java problem for you to try**

*A parking garage charges****$10****minimum fee to park for up to three hours. The garage charges an additional****$2****per hour or part thereof in excess of three hours. The maximum charge for any given 24-hour period is****$50****. Assume that no car parks for longer than 24 hours at a time.*

*Write a Java application that calculates and displays the parking charges for each customer who parked in the garage the previous day. You should enter in a dialog box the hours parked for each customer. The program calculates and displays the running total of previous day’s receipts. The program should use the method* ***calculateCharges()****to determine the charge for each customer.*

Well, that is all we have for Java classes but I would recommend that you practice more and look for more resources about **Java classes** and **methods**.