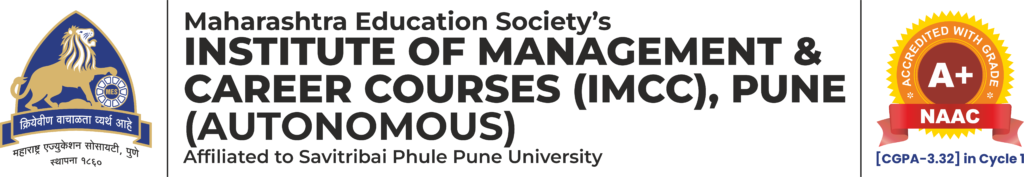
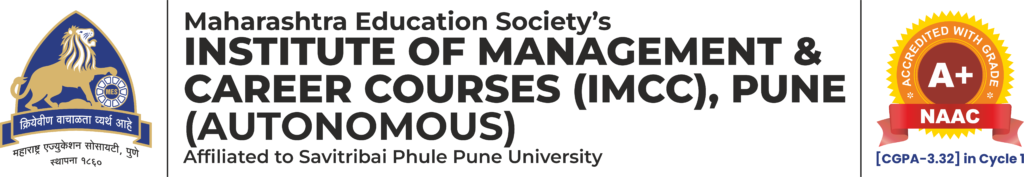


**IT21 - Java Programming**   
**Practical Assignments**   
**A.Y 2024-25**

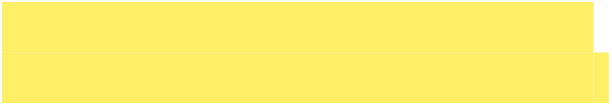
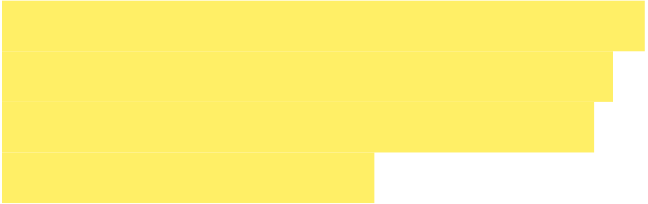
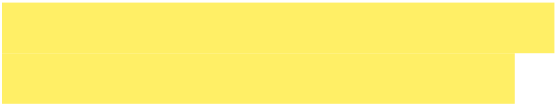
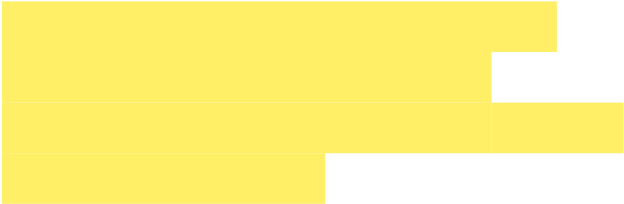
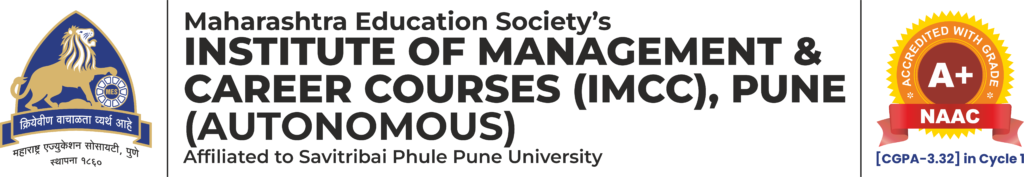
|  |  |  |
| --- | --- | --- |
| **Sr Nbr** | **Assignment** | **Expected Output** |
| 1 | In the project ManageEmployees, add a package  pkgOrgStructre. Within that package add classes Employee, Manager,Developer and Tester and EmployeeDemo. All classes except EmployeeDemo should have its constructor.  Connect the valid business classes using **Inheritance**. | |  |  | | --- | --- | | ​ | Proper workspace, project, |   package and classes should get   |  |  |  | | --- | --- | --- | |  | |  | | --- | | created using Eclipse. | | | |  | | --- | | ​ | | Correct use of Inheritance |   should be demonstrated while linking the classes in IS-A  hierarchy.   |  |  | | --- | --- | | ​ | Every business class should |   have a constructor.   |  |  | | --- | --- | | ​ | main() should be added to the |   Demo class using proper syntax |
| |  | | --- | | 2 | | In the classes created above make the following changes. ​a. Add the attributes empId, empNm, empAge, empPanCrd, empAadharCrd, empDept, empSal, empContactNbr,empExp in Employee class. ​ b. Add the attributes, noOfTeams, noOfReportees in the Manager class. ​ c. Add the attributes, nameOfTheTeam, nameOfManager, technologies in the Developer class.​ d. Add the attributes, nameOfTheTeam, nameOfManager, testingType in the Tester class. ​ e. Demonstrate **Encapsulation** in each of these classes. Also add **accessors and mutators.**​ f. Modify the constructor to initialise the instance variables using appropriate setter methods.​ g. Add main() to EmployeeDemo class. Create objects of Employee, Manager, Developer and Tester classes using correct initialisation values. | |  |  | | --- | --- | | ​ | Correct attributes should be |   added to respective classes.   |  |  | | --- | --- | | ​ | Encapsulation should be |  |  | | --- | | achieved using correct access |  |  |  | | --- | --- | |  | specifier for instance variables | | ​ | Getter setter (accessor |  |  | | --- | | mutators) should be defined |   using right syntax. And used   |  |  | | --- | --- | |  | correctly in the constructor. | | ​ | Objects of the classes should |   be created using correct syntax. |
| |  | | --- | | 3 | | In the classes created above make the following changes:​a. Add details() in the Employee class. It should print the values of all the instance variables of the Employee class.​b. Add computeSalary() in the Employee class. This should be used by the subclasses to calculate their salary.​ | |  | | --- | | ​ | |



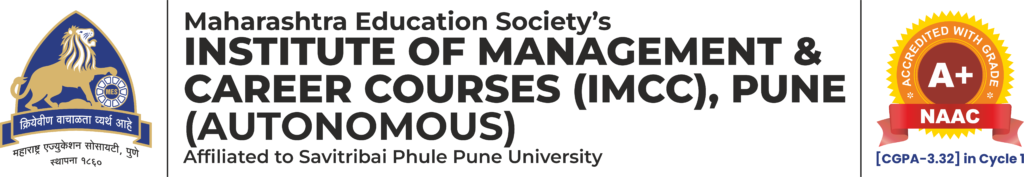
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | |  |
|  | c. **Override** details() in the sub classes such that it prints the values of the instance variables of that respective class. ​d. **Override** computeSalary() in the sub classes such that it calculates the salary of a given Employee and sets it. For a Manager the salary formula is, salary\*8​ For Developer the salary formula is, salary\*6​ For Tester the salary formula is, salary\*5 | |  | |
| 4 | In the previously designed classes, demonstrate use of **super keyword** to invoke super class details() from within its sub   |  | | --- | | classes. | | |  | |
| 5 | Create a package “InterfaceAbstractpkg”. In this package write an interface Area having methods compute() and print(). It also has a constant for Pi. This interface is used to compute area of given shape. Write Rectangle class and Circle class which implement this interface and override its methods. Write another class having objects of Rectangle class and Circle class and invoke their respective methods. | |  | |
| 6 | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Write | an | interface | Employee | having | methods | | viz, | | enroll(),terminate(),calculatePay(). | | | | Write | | classes | |   ProductionDepartment, EngineeringDepartment to implement Employee interface and override its methods. enroll() method should take details from the employee and add him / her to the respective department and print a confirmatory message. terminate() should remove an employee from the department with a print statement. calculatePay() should print the total salary of the employee depending on his/her department and pay scale of that department. Create objects of these classes and invoke overridden methods on those objects. | |  | |
| 7 | You are running a shared cab service. Write a class, BookARide. In this class write the method, bookSeat(int totalNoOfBooking) In bookSeat() if totalNoOfBooking exceeds 4 throw RideOverBooked exception. Write your own user defined exception class, RideOverBooked. | |  | |
| 8 | Define an Account class with variables acc\_no, acc\_type, name and balance. Write a program to accept data of 10 account holders. Save these objects using collection framework class of your choice. Print names of customers having balance greater than 10,000 | |  | |
| 9 | Write an interface, TemperatureConverter. Add methods, convertToFarenheit() and onvertToCelsius(). Add a static method about() which will print the purpose of this interface.   |  |  |  |  | | --- | --- | --- | --- | | Implement | the | interface | into |   TemperatureConverterImplementation class and override the necessary methods. Create objects of the business class in TemperatureConverterDemo class and demonstrate the behaviour of all the methods. | |  | |



|  |  |  |
| --- | --- | --- |
| 10 | Write a Java programming to create a banking system with three classes - Bank, Account, SavingsAccount, and  CurrentAccount. The bank should have a list of accounts and methods for adding them. Accounts should be an interface with methods to deposit, withdraw, calculate interest, and view balances. SavingsAccount and CurrentAccount should implement the Account interface and have their own unique methods. |  |
| 11 | Write an interface, SortingUtility having methods,  ascendingSort() and descendingSort(). Implement this in SortArray class and SortString class which will inturn  override the two abstract methods in the interface. Create necessary demo class and call relevant methods on the objects​ | I/P : If the string input is Apple then the sorted ascending output should be Aelpp and descending output  should be ppleA |
| 12 | Write a Java program to create an interface Encryptable with methods encrypt (String data) and decrypt (String  encryptedData) that define encryption and decryption  operations. Create two classes AES and RSA that implement the Encryptable interface and provide their own encryption and decryption algorithms. |  |
| 13 | You have developed an e-commerce website for your client.  The maximum no of units of a single prodcut that one user can add to the cart is 5. If the user adds more than 5 units of a single product, then your application is expected to throw, **MaximumProductsLimitExceededException**. Write a custom exception class to achieve this. |  |
| 14 | The manufacturing of your medical company has very strict standards of product specifications. After each pill /tablet is ready, it is weighed. If the weight of the tablet exceeds the allowed limit, **TabletWeightExceededException** is raised.  Using exception handling in Java, write the program to achieve the above business requirement. |  |
| 15 | When the battery of your mobile phone is less than 20%, the system should generate, **LowBatteryException** to alert the user to start charging the device. If the battery goes lower than 10 then the system should raise  InsufficientChargeException and put the unit on power saver mode. Using exception handling in Java, write the program to achieve the above business requirement |  |
| 16 | You are writing an app for taking names of the volunteers for Cultural Committee of your Institute. According to the  guidelines only 15 members are allowed in the committee. Using your app, take the names of the interested candidates till the number reaches 15. Once the threshold is crossed, display a message, “No more candidates allowed as  volunteers. Thank you”. Use ArrayList to achieve the above given business logic. | Hint : You will have to keep  checking the size of the arraylist. |



|  |  |  |
| --- | --- | --- |
| 17 | Once the above list of volunteers is finalized, each volunteer needs to pick a historical character as his/her badge icon. Using your app, take the name of the historical character from the volunteers and store them for future uses. Also, no two characters should be the same. In case the character is already in the list, ask the volunteer to enter some other character. Use ArrayList to achieve the above given requirement. | Hint : You will have to check if the element is already contained inside the list. |
| 18 | The placement cell of your Institute has asked you to share the name of one technology which you are expert in. Using an app, take this from 15 students. The cell then wants you to give a technology count based on the input. For ex, how many students chose Java, how many chose Python, how many entered MERN, etc. Demonstrate the use of ArrayList to achieve this. | Hint : You will need to sort the  arraylist and then count the  individual elements pertaining to a given technology. |
| 19 | For the recently held HR meet, the CR and LR of division A and B marked the attendance for their respective classes in separate lists. The TPO cell wants a consolidated list of FYMCA students who were present for the event. Write a program using ArrayList to mark division wise attendance first and then give the consolidated list. | Hint : You will need to add the arraylists to get the final one. |
| 20 | Sports cell of the Institute needs to choose its core team from those students who participated in the recently held sports events. For this, the sports coordinator has decided to consider the participants of Football and Cricket. Only those players who participated in BOTH these games will be considered for the core team. Using ArrayList, write a Java program which will take the names of the students participating in Football and Cricket. Find the common names in these two events and put them into a third list, called, SportsCoreTeam | Hint : You will need to compare the two arraylists to get the third one |
| 21 | The top three scorers in the coding competition will be given a certificate and trophy by the Coding Club. Using Vector, take the final scores (out of 100) of the participating coders and find the top three using only the max() function available   |  | | --- | | in Collections. | | Hint : There is no need to sort the vector. |
| 22 | KKR and MumbaiIndians are going to play the kickstart match of this year’s IPL season. Using Vector, you have taken the names of the players in each team and are displaying the same. But there is a last minute change in the batting line up of KKR. In place of QuintonDeKock, the team will send Anukul Roy at two down position. Make this change in their batting line up and display the new order. | |  | | --- | | Hint : You need to get the original |   element and set it with the new one. |
| 23 | In the e-commerce portal designed by you, the customer adds products to the shopping cart. Use a vector to hold the objects of Product class. At the time of billing, access each product object and read its price. Add the cost of all the products and display the bill total. If the cart is empty, show a message, “Can we help you in finding what you were looking for?” and end the billing process. | Hint : You will need a Product class and its objects. You will also need to check if the vector holding the cart is empty or not? |
| 24 | During the Marathon event the organisers maintained a list to hold the details of the finishers. Once the marathon got over, they displayed the details of the first runner to finish the | Hint : You will need to check the first and last element in the vector. |



|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
|  | marathon and the last one to finish the same. Write an app having the objects of MarathonRunner class in to a vector list, finishers. Display the details of the runner who comes first and of the who comes last. MarathonRunner class has the properties, name, badgeNbr, startTime and endTime |  | |