|  |  |  |
| --- | --- | --- |
|  | | |
| Jjjj | **COURSE OUTLINE**  nn | |
| 1 | School | Faculty of Science and Engineering (FSE) |
| 2 | Department | CSE |
| 3 | Programme | B.Sc. in CSE |
| **4** | **Name of Course** | Object Oriented Programming |
| **5** | **Course Code** | CSE 201 |
| **6** | **Trimester** | Spring, 2021 |
| **7** | **Pre-requisites** | CSE 103 |
| **8** | **Status** | Core CSE Course |
| **9** | **Credit Hours** | 3 |
| **10** | **Section** | 193EA |
| **11** | **Class Hours** | |  |  |  |  | | --- | --- | --- | --- | | **Section** | **Class Day** | **Class Hours** | **Venue** | | 193EA | Friday | 08:00 AM-09:15 AM | Online and (B-402) | | Friday | 09:15 AM-10:30 AM | Online and (B-402) | |
| **12** | **Class Location** | City Campus |
| **13** | **Course website** | <https://classroom.google.com/u/1/c/MjYyNTE4NjIxMjM5>  Code: yfzjhmw |
| **14** | **Instructor** | Dr. Muhammad Aminur Rahaman |
| **15** | **Contact** | [aminur@cse.green.edu.bd](mailto:%20aminur@cse.green.edu.bd) |
| **16** | **Office** | Room No: B-517 |
| **17** | **Counselling Hours** | |  |  |  | | --- | --- | --- | | **Day** | **Counseling Hours** | **Venue** | | Mon | 03:00pm-06:00pm | Room No B-517 (Zoom Online) | | Wed | 06:00pm-08:30pm | Room No B-517 (Zoom Online) | | Fri | 03:30pm-06:00pm | Room No B-517 (Zoom Online) | |
| **18** | **Text Book** | 1. Java-The Complete Reference (7th Edition) by Herbert Schildt 2. Introduction to Java Programming, Comprehensive Version, Y. Daniel Liang, (10th Ed.) 3. Deitel P.J., Deitel H.M. - Java How to Program, 9th Edition 4. Head First Java by Kathy Sierra & Bert Bates, 2nd Edition |
| **19** | **Reference** | 1. Tutorial on GUI development step by step 2. [Stack Overflow - Where Developers Learn, Share, & Build Careers](https://stackoverflow.com/) |
| **20** | **Equipment & Aids** | The course website provides links to a number of online resources to assist students in learning programming in OOP. A number of related online text books and projects will be found in the course websites for helping student. |
| **21** | **Course Rationale** | Object-oriented programming (OOP) is a programming language model organized around objects rather than "actions" and data rather than logic. Historically, a program has been viewed as a logical procedure that takes input data, processes and produces output data. The programming challenge was seen as how to write the logic, not how to define the data. Object-oriented programming takes of what we really care about are the objects we want to manipulate rather than the logic required to manipulate them. |
| **22** | **Course Description** | This course is designed as an entry level programming course for students who have prior programming experience. This course introduces the concepts of object-oriented programming to students with a background in the procedural paradigm. The course begins with a brief review of control structures and data types with emphasis on structured data types and array processing. It then moves on to introduce the object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design. Brief review of control structures, functions, and primitive data types - Object-oriented programming: Object-oriented design; encapsulation and information-hiding; separation of behavior and implementation; classes, subclasses, and inheritance; polymorphism; class hierarchies, introduction to Graphical User Interface (GUI): preparing simple graphics APIs and Games. |
| **23** | **Course Outcomes (CO)** | On successful completion of this course students will be able to-  CO1: Explain the basic concepts of object-oriented programming language, syntax, class, object, inheritance, interface and exception handling etc.  CO2: Develop codes in object-oriented programming language to solve simple and moderately complex problems.  CO3: Design an application-based software or to complex problems using the advanced features and tools of object-oriented programming. |
| **24** | **Teaching Methods** | This course has four contact activities: lectures, practical, presentation and small group discovery. Each of these activities is used to support interactive learning.  Lectures will be used primarily to introduce the core theoretical material of the course. The practical is focused on developing core programming skills. The presentation is the collaborative sessions with groups and individual student sharing course related topic and activity. The small group discovery sessions focus on a guided research experience in an area of ethics relevant to Software Engineering. The small groups will do most of their work outside these sessions. |
| **25** | **Topic Outline**  All topics and problems are from the main text if not specified otherwise. | |
|  | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Lecture** | **Selected Topics** | **Article**  **(Text)** | **Suggested Problems.**  **(Text)** | **Outcome** | |  | | | | | | (1) | Introducing OOP | 1.1 | 1.1, 1.3, 2.1, 3.3, 5.1, 5.5 | CO1 | | OOP vs SPL | 2.1 | | Data Types, Variables, Operators | 3.1, 4.1 | | Branching Statements: If-else, Switch | 5.1 | |  | | | | | | (2) | Iterative Statements | 5.2 | 5.7, 5.8, 5.9, 5.10 | CO1 | | While, do-while, for and for each | 5.2.1-5.2.4 | | Nested loop | 5.2.5 | | Jumping statement; break, level break, continue | 5.3 | |  | | | | | | (3) | Introducing Arrays | 3.11 | 3.15, 3.16, 3.21 | CO1, CO2 | | Multi-dimensional Arrays | 3.12 | | Sorting | 3.13 | | Searching | 3.14 | |  | | | | | | (4) | Introducing classes, objects | 6.1 | 6.3, 6.5, 6.7, 6.9 | CO1, CO2 | | Encapsulation and Constructors | 6.4 | | Instance variable hiding problem and this keyword | 6.5 | | Garbage collection | 6.6 | |  | | | | | | (5-7) | Method overloading | 7.1 | 7.1, 7.3,7.5, 7.7, 7.9, 7.11, 7.13, 7.15 | CO1, CO3 | | Overloading constructors | 7.2 | | Using object as parameters and returning object | 7.3 | | Recursion | 7.4 | | Introducing access control | 7.5 | | Understanding static and final | 7.6 | | Nested classes | 7.7 | |  | | | | | | (8-11) | Introducing inheritance | 8.1 | 8.1, 8.3, 8.5, 8.7, 8.9, 8.11 | CO2, CO4 | | Use of super | 8.2 | | Creating multi-level hierarchy | 8.3 | | Method overriding | 8.4 | | Abstract classes | 8.5 | | Use of final in class and methods | 8.6 | | Using library class | 8.7 | |  | | | | | | (12-14) | Introducing package | 9.1 | 9.3, 9.4, 9.6 | CO1, CO4 | | Access protection | 9.2 | | Interfaces | 9.3 | | Interfaces vs classes | 9.4 | |  | | | | | | (15-16) | Introducing exception handling | 10.1 | 10.3, 10.6, 10.8, 10.9 | CO1, CO4 | | Exception type, exception catching | 10.2 | | Try-catch, throw, throws, finally | 10.3-10.6 | | Creating own exception | 10.8 | |  | | | | | | (17-19) | Introducing multithreaded programming | 11.1 | 11.3, 11.5, 11.7, 11.9 | CO1, CO2 | | Creating thread | 11.2-11.4 | | Synchronization | 11.5 | | Interthread communication | 11.6 | |  | | | | | | (20-21) | Introducing String class | 15.1 | 15.1, 15.5, 15.8, String addition, subtraction and multiplication | CO1, CO3 | | String operation | 15.2-15.5 | | Big number manipulation using String | Reference topics | |  | | | | | | (22-24) | Introducing AWT and its component | 23.1-23.3 | 23.5, 23.9, 24.5, 29.6, 29.8, Project Presentation | CO2, CO3, CO4 | | Graphical drawing | 23.6-23.8 | | Animation | 24.5 | | Introducing Swing and its component | 29.1-29.3 | | Simple software interface design | 29.4-29.5 | | Database Connection | Reference topics | |  | | | | | | |
| **26** | **Assessment and Marks Distribution:** | Students will be assessed on the basis of their overall performance in all the exams, quizzes, and class participation. Final numeric reward will be the compilation of (tentative):   * Class Tests (15%) * Assignment (5%) * Individual Presentation (5%) * Attendance and Performance (5%) * Midterm Examination (30%) * Final Examination (40%) |
| **27** | **Assessment Methods of Cos** | Assessment methods of COs are given below:   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  | **Assessment** | | | | | | | | **COs** | **CT1** | **CT2** | **CT3** | **MT** | **FE** | **Assignment** | **Presentation** | | **CO1** | **√** |  |  | **√** | **√** |  |  | | **CO2** |  | **√** |  | **√** | **√** | **√** |  | | **CO3** |  |  | **√** | **√** | **√** | **√** | **√** | |
| **28** | **Mapping of COs with POs** | Mapping of COs with program outcomes (POs) are given below:   |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Program Outcomes (POs)** | | | | | | | | | | | | | | **COs** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | | **CO1** | **√** |  |  |  |  |  |  |  |  |  |  |  | | **CO2** |  | **√** |  |  |  |  |  |  |  |  |  |  | | **CO3** |  |  | **√** |  |  |  |  |  |  |  |  |  | |
| **29** | **Grading Policy** | The following chart will be followed for grading. This has been customized from the guideline provided by the School of Engineering and Computer Science.   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **A+** | **A** | **A-** | **B+** | **B** | **B-** | **C+** | **C** | **D** | **F** | | **80 and above** | **75-<80** | **70-<75** | **65-<70** | **60-<65** | **55-<60** | **50-<55** | **45-<50** | **40-<45** | **<40** | |
| **29** | **Additional Course Policies** | |  |  | | --- | --- | | Assignments | There will be four assignments. Average marks of the assignments will be counted. No late homework will be accepted.  ***Any kind of copy in assignment will carry zero mark.***  Two or more copied assignments will carry zero mark in all assignments. Zero tolerance will be shown in this regard. Solutions to assignment problems will be provided through web and on hand. | | Class Test | There will be at least three CTs, best of two will be counted. A CT can be taken with an announcement in prior or without any announcement. | | Exams | Midterm and final examination will be closed book, closed notes. Mobile is strictly prohibited in exam hall. Please bring your own watch and synchronize time during exam hours. | | Test Policy: | If you are absent from a test, and you have not spoken to the teacher personally beforehand, your grade for the test will be zero. No make-up for class test will be taken because it has alternative (three out of four). No make-up for mid will be entertained without presence and recommendation of guardian and written permission of the department. Make-up test of mid will be much harder than the regular test. | |
| **30** | **Additional Information** | 1. Academic Calendar Fall 2019: http://www.green.edu.bd/academics/academic-calendar. 2. Academic Information and Policies: http://www.green.edu.bd/academics/academic-rules-a-regulations. 3. Grading and Performance Evaluation: http://www.green.edu.bd/academics/academic-rules-a-regulations. 4. Proctorial Rules: http://www.green.edu.bd/administrator/proctors-office. |