|  |  |
| --- | --- |
| **Jan, 2019 Date** | **Topics** |
| **1Jan** | **Introduction of O.S.** |
| **2 Jan** | **Continue** |
| **3 Jan** | **Memory mgt. Of Single User v/s Multi User** |
| **4 Jan** | **Architecture** |
| **5 Jan** | **Continue** |
| **6 Jan** | **Sun day** |
| **7 Jan** | **Functions** |
| **8 Jan** | **Characteristics** |
| **9 Jan** | **Historical Evolution** |
| **10 Jan** | **Types of operating system** |
| **11 Jan** | **Continue** |
| **12 Jan** | **Operating System** |
| **13 Jan** | **Sunday** |
| **14 Jan** | **HOLIDAY** |
| **15 Jan** | **Distributed and Parallel** |
| **16 Jan** | **Continue** |
| **17 Jan** | **O.S. as resource Manager** |
| **18 Jan** | **Continue** |
| **19 Jan** | **partitioning** |
| **20 Jan** | **Sunday** |
| **21 Jan** | **I/O structure** |
| **22 Jan** | **Storage structure** |
| **23 Jan** | **Continue** |
| **24 Jan** | **swapping** |
| **25 Jan** | **Storage Hierarchy** |
| **26 Jan** | **REPUBLIC DAY** |
| **27 Jan** | **paging** |
| **28 Jan** | **System Components** |
| **29 Jan** | **Continue** |
| **30 Jan** | **Services** |
| **31 Jan** | **Continue** |

**SUBJECT:-COMPUTER (OPERATING SYSTEM)**

**Name of Assistant Prof: DIKSHA AGGARWAL**

|  |  |
| --- | --- |
| **1 Feb** | **System Calls** |
| **2 Feb** | **System Programs** |
| **3 Feb** | **Sunday** |
| **4 Feb** | **Continue** |
| **5 Feb** | **System Structure** |
| **6 Feb** | **Continue** |
| **7 Feb** | **Continue** |
| **8 Feb** | **Continue** |
| **9 Feb** | **Process Mgt.** |
| **10 Feb** | **Sunday** |
| **11 Feb** | **Continue** |
| **12 Feb** | **Process Concepts** |
| **13 Feb** | **Continue** |
| **14 Feb** | **Process State** |
| **15 Feb** | **Continue** |
| **16 Feb** | **Process Control Block** |
| **17 Feb** | **Sunday** |
| **18 Feb** | **continue** |
| **19 Feb** | **HOLIDAY** |
| **20 Feb** | **continue** |
| **21 Feb** | **Operations** |
| **22 Feb** | **Continue** |
| **23 Feb** | **Continue** |
| **24 Feb** | **Sunday** |
| **25 Feb** | **Process scheduling** |
| **26 Feb** | **Continue** |
| **27 Feb** | **Continue** |
| **28 Feb** | **HOLIDAY** |
| **March, 2019** | |
| **1March** | **Inter-process Communication** |
| **2 March** | **continue** |
| **3 March** | **Sunday** |
| **4 March** | **HOLIDAY** |
| **5 March** | **Continue** |
| **6 March** | **CPU scheduling** |
| **7 March** | **Scheduling Criteria** |
| **8 March** | **Continue** |
| **9 March** | **Continue** |
| **10 March** | **Sunday** |
| **11 March** | **Levels of Scheduling** |
| **12 March** | **Continue** |
| **13 March** | **Continue** |
| **14 March** | **Scheduling Algorithms** |
| **15 March** | **Continue** |
| **16 March** | **Continue** |
| **17 March** | **Sunday** |
| **18 March -24march** | **Vacations** |

|  |  |
| --- | --- |
| **25 March** | **Deadlocks** |
| **26 March** | **charcterization** |
| **27 March** | **Methods of handling** |
| **28 March** | **Continue** |
| **29 March** | **Deadlock detection** |
| **30 March** | **Prevention** |
| **31 March** | **Sunday** |
| **April, 2019** | |
| **1 April** | **Avoidance** |
| **2April** | **Continue** |
| **3 April** | **Recovery** |
| **4 April** | **Disk scheduling, disk structure, disk Management** |
| **5 April** | **File systems- Functions, file access and allocation methods** |
| **6 April** | **Continue** |
| **7 April** | **Sunday** |
| **8 April** | **Continue** |
| **9 April** | **Continue** |
| **10 April** | **Revision** |
| **11 April** | **Virtual memory & Page replacement method** |
| **12 April** | **Continue** |
| **13 April** | **Revision** |
| **14 April** | **Sunday** |
| **15 April** | **Critical section problem** |
| **16 April** | **continue** |

|  |  |
| --- | --- |
| **Jan, 2019 Date** | **Topics** |
| **1Jan** | **OBJECT ORIENTED PROGRAMMING** |
| **2 Jan** | **Features & Benefits** |
| **3 Jan** | **Features of OOP with C++** |
| **4 Jan** | **Class & Objects** |
| **5 Jan** | **Continue** |
| **6 Jan** | **Sunday** |
| **7 Jan** | **Revision** |
| **8 Jan** | **Date hiding & Encapsulation** |
| **9 Jan** | **Structure** |
| **10 Jan** | **Data Members & Member functions** |
| **11 Jan** | **Rivision** |
| **12 Jan** | **Rivision** |
| **13 Jan** | **Sunday** |
| **14 Jan** | **HOLIDAY** |
| **15 Jan** | **Scope Resolution operator & its significance** |
| **16 Jan** | **Continue** |
| **17 Jan** | **Continue** |
| **18 Jan** | **Static data Members** |
| **19 Jan** | **Revision** |
| **20 Jan** | **Sunday** |
| **21 Jan** | **Static member function** |
| **22 Jan** | **Nested & Local Class** |
| **23 Jan** | **Continue** |
| **24 Jan** | **Accessing member of class & Structures** |
| **25 Jan** | **Continue** |
| **26 Jan** | **REPUBLIC DAY** |
| **27 Jan** | **Sunday** |
| **28 Jan** | **Continue** |
| **29 Jan** | **Initialization of constructor** |
| **30 Jan** | **Types of constructor** |
| **31 Jan** | **Continue** |

**SUBJECT : COMPUTER (OBJECT ORIENTED PRAGRAMMING WITH C++)**

**Name of Assistant Prof: DIKSHA AGGARWAL**

|  |  |
| --- | --- |
| **1 Feb** | **Continue** |
| **2 Feb** | **Continue** |
| **3 Feb** | **Sunday** |
| **4 Feb** | **Constructor overloading** |
| **5 Feb** | **Default values to Parameter** |
| **6 Feb** | **Continue** |
| **7 Feb** | **Continue** |
| **8 Feb** | **Destructor** |
| **9 Feb** | **Revision** |
| **10 Feb** | **Sunday** |
| **11 Feb** | **Console I/O** |
| **12 Feb** | **Continue** |
| **13 Feb** | **Continue** |
| **14 Feb** | **Formatted & Unformatted I/O operations** |
| **15 Feb** | **Continue** |
| **16 Feb** | **Continue** |
| **17 Feb** | **Sunday** |
| **18 Feb** | **Revision** |
| **19 Feb** | **HOLIDAY** |
| **20 Feb** | **Manipulators** |
| **21 Feb** | **Continue** |
| **22 Feb** | **Friend Function** |
| **23 Feb** | **Friend Class** |
| **24 Feb** | **Sunday** |
| **25 Feb** | **Continue** |
| **26 Feb** | **Arrays, arrays of Objects** |
| **27 Feb** | **Passing & Returning Objects to Functions** |
| **28 Feb** | **HOLIDAY** |
| **1March** | **Continue** |
| **2 March** | **Continue** |
| **3 March** | **Sunday** |
| **4 March** | **HOLIDAY** |
| **5 March** | **Revision** |
| **6 March** | **String handling in C++, Pointers** |
| **7 March** | **Continue** |
| **8 March** | **New & Delete Operator** |
| **9 March** | **Continue** |
| **10 March** | **Sunday** |
| **11 March** | **Array of Pointers to Objects** |
| **12 March** | **Continue** |
| **13 March** | **This Pointer** |
| **14 March** | **Continue** |
| **15 March** | **Passing parameter to to function by reference & Pointer** |
| **16 March** | **Continue** |
| **17 March** | **Sunday** |
| **18 March -24march** | **Vacations** |

|  |  |
| --- | --- |
| **25 March** | **Revision** |
| **26 March** | **Continue** |
| **27 March** | **Continue** |
| **28 March** | **Revision** |
| **29 March** | **Static Polymorphism- operators in C++** |
| **30 March** | **Continue** |
| **31 March** | **Sunday** |
| **1 April** | **Precedence & Associativity Rules** |
| **2April** | **Continue** |
| **3 April** | **Continue** |
| **4 April** | **Operator Overloading** |
| **5 April** | **Continue** |
| **6 April** | **Continue** |
| **7 April** | **Sunday** |
| **8 April** | **Unary & Binary operators overloading** |
| **9 April** | **Continue** |
| **10 April** | **Revision** |
| **11 April** | **Function overloading** |
| **12 April** | **Revision inline Functions** |
| **13 April** | **Continue** |
| **14 April** | **Sunday** |
| **15 April** | **Merits/Demerits of static Polymorphism** |
| **16 April** | **continue** |

**Dr. Mamta Sharma**

**(Principal)**