**“Obyekt-yönlü proqramlaşdırma (C++)” fənni üzrə**

**TƏDRİS-TEMATİK PLAN**

| **№** | **Tarix** | **Mövzular** | **Tədris materialının məzmunu və sərbəst işlər** | | | **Saat** | **Qeyd** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **I BÖLMƏ. Obyekt-yönlü proqramlaşdırmanın konsepsiyaları (Object-Oriented Programming Concepts).** | | | | | | | |
|  |  | **Introduction to Object Oriented Programming (OOP).** | | | **Giriş. Obyekt-yönlü proqramlaşdırma (OYP) fənninin predmeti, məqsəd və vəzifələri.** |  |  |
|  |  | **Machine Languages, Assembly Languages and High-Level Languages.** | | |  |  |  |
|  |  | **History of C and C++. C++ Standard Library.** | | |  |  |  |
|  |  | **Introduction to C++ Programming.** | | |  |  |  |
|  |  | **First Program in C++: Printing a Line of Text. Modifying Our First C++ Program.** | | |  |  |  |
|  |  | **Another C++ Program: Adding Integers. Memory Concepts. Arithmetic.** | | |  |  |  |
|  |  | **Decision Making: Equality and Relational Operators.** | | |  |  |  |
|  |  | **Basic Principles of Object Oriented Programming.** | | | **Obyekt-yönlü proqramlaşdırmanın əsas prinsipləri, mahiyyəti.** |  |  |
|  |  | **Issues with Procedure Oriented Programming.** | | | **Prosedur-yönlü proqramlaşdırmanın problemləri.** |  |  |
|  |  | **Procedure Oriented versus Object Oriented Programming.** | | | **Prosedur-yönlü və obyekt-yönlü proqramlaşdırmanın müqayisəsi.** |  |  |
|  |  | **Basic of Object Oriented Programming.** | | | **Obyekt-yönlü proqramlaşdırmanın əsasları.** |  |  |
|  |  | **Concept of Object Oriented Programming:**   * **Object** * **Class** * **Abstraction** * **Encapsulation** * **Inheritance** * **Polymorphism** | | | **Obyekt-yönlü proqramlaşdırma konsepsiyaları:**   * **Obyekt** * **Sinif** * **Abstraksiya** * **İnkapsulyasiya** * **Varislik** * **Polumorfizm** |  |  |
|  |  | **Example of Some Object Oriented Languages.** | | | **Obyekt-yönlü proqramlaşdırma dilləri nümunələri.** |  |  |
|  |  | **Advantages and Disadvantages of OOP.** | | | **OYP-nin üstünlükləri və çatışmazlıqlar.** |  |  |
|  |  | **Questions and Exercises:**   * **What Is an Object?** * **What Is a Class?** * **What Is Inheritance?** * **What Is an Interface?** * **What Is a Package?** | | | **Suallar və tapşırıqlar:**   * **Obyekt nədir?** * **Sinif nədir?** * **Varislik nədir?** * **İnterfeys nədir?** * **Paket nədir?** |  |  |
|  |  | **Introduction to C++:**   * **The Need of C++** * **Features of C++** * **C++ Versus C** * **History of C++** | | | **C++ proqramlaşdırmaya giriş:**   * **C++ proqramlaşdırmaya yaranan ehtiyaclar** * **C++ proqramlaşdırmanın özəllikləri** * **C++ və C proqramlaşdırma** * **C++ proqramlaşdırmanın tarixi** |  |  |
| **II BÖLMƏ. C++ Proqramlaşdırma dilinin konstruksiyaları (C++ Language Constructs).** | | | | | | | |
|  |  | **Object-oriented programming systems. Classification of object-oriented programming systems, General principles of operation. Summary of object-oriented programming systems.** | | **Obyekt-yönlü proqramlaşdırma sistemləri. Obyekt-yönlü proqramlaşdırma sistemlərinin təsnifatı, ümumi iş prinsipləri. Obyekt-yönlü proqramlaşdırma sistemlərinin xülasəsi.** | |  |  |
|  |  | **C++ Program Structure.** | | **C++ proqramlarının strukturu.** | |  |  |
|  |  | **Desing Object Oriented Programming.** | | **Obyekt-yönlü proqramların layihələndirilməsi.** | |  |  |
|  |  | **Character Set and Tokens:**   * **Keywords** * **Identifiers** * **Literals** * **Operators and Punctuators** | | **Simvollar və işarələr yığımı:**   * **Açar sözlər** * **İdentifikatorlar** * **Literallar** * **Operatorlar və durğu işarələri** | |  |  |
|  |  | **Variable Declaration and Expression.** | | **Dəyişənlərin təsviri və ifadə olunması.** | |  |  |
|  |  | **Statements.** | | **İfadələr.** | |  |  |
|  |  | **Data Type.** | | **Verilənlərin tipləri.** | |  |  |
|  |  | **Type Conversion and Promotion Rules.** | | **Tiplərin təyini və çevrilməsi qaydaları.** | |  |  |
|  |  | **Preprocessor Directives.** | | **Preprosessorun direktivləri.** | |  |  |
|  |  | **Namespace.** | | **Əhatə dairəsi.** | |  |  |
|  |  | **User Defined Constant const.** | | **İstifadəçi tərəfindən təyin olunan sabitlərin tipi – const.** | |  |  |
|  |  | **Input/Output Streams and Manipulators.** | | **Giriş / Çıxış axınları və manipulyatorları.** | |  |  |
|  |  | **Dynamic Memory Allocation with new and delete.** | | **Yaddaşın dinamik idarə olunması – new və delete komandası.** | |  |  |
|  |  | **Condition and Looping.** | | **Durum və döngü – dövrlər.** | |  |  |
|  |  | **Functions**   * **Function Syntax** * **Function Overloading** * **Inline Functions** * **Default Argument** * **Pass by Reference** * **Return by Reference** | | **Funksiyalar**   * **Funksiyaların sintaksisi** * **Həddini aşma funksiyası** * **Sətirarası funksiyalar** * **Susma halında arqumentlər** * **İstinadlar üzrə ötürmələr** * **İstinadlar üzrə qayıdışlar** | |  |  |
|  |  | **Array, Pointer and String.**  **Structure, Union and Enumeration.** | | **Massivlər, Göstəricilər və Sətirlər.**  **Strukturlar, Birləşmələr və Sadalamalar.** | |  |  |
| **III BÖLMƏ. Obyektlər və siniflər (Objects and Classes).** | | | | | | | |
|  |  | **Introduction to Classes and Objects.** | |  | |  |  |
|  |  | **Classes, Objects, Member Functions and Data Members.** | |  | |  |  |
|  |  | **Defining a Class with a Member Function.** | |  | |  |  |
|  |  | **Defining a Member Function with a Parameter.** | |  | |  |  |
|  |  | **Data Members, set Functions and get Functions.** | |  | |  |  |
|  |  | **Initializing Objects with Constructors.** | |  | |  |  |
|  |  | **Placing a Class in a Separate File for Reusability.** | |  | |  |  |
|  |  | **Separating Interface from Implementation.** | |  | |  |  |
|  |  | **Validating Data with set Functions.** | |  | |  |  |
|  |  | **(Optional) Software Engineering Case Study: Identifying the Classes in the ATM Requirements Document** | |  | |  |  |
|  |  | **Purpose of objects. Types of objects.** | | **Obyektlərin təyin edilməsi. Obyektlərin tipləri.** | |  |  |
|  |  | **Access to the fields of the object. Access Specifiers. Data transmission.** | | **Obyekt sahələrinə müraciət. Müraciət spesifikatorları. Məlumatın ötürülməsi.** | |  |  |
|  |  | **Objects and the Member Access.** | | **Obyektlər və onlara əlyetərlik.** | |  |  |
|  |  | **C++ Classes.** | | **C++ Sinifləri.** | |  |  |
|  |  | **Class, instance, and inheritance.** | | **Sinif, nüsxə və varislik.** | |  |  |
|  |  | **Classes and methods. Types of classes.** | | **Siniflər və metodlar. Siniflərin növləri.** | |  |  |
|  |  | **Defining Member Function.** | | **Üzv Funksiyasının təyini.** | |  |  |
|  |  | **Constructor.**   * **Default Constructor.** * **Parameterized Constructor.** * **Copy Constructor.** | | **Konstruktor.**   * **Susma halında konstruktor.** * **Parametrik konstruktor.** * **Konstruktor surəti.** | |  |  |
|  |  | **Destructors.** | | **Destruktorlar.** | |  |  |
|  |  | **Object as Function Arguments and Return Type.** | | **Obyekt funksiya arqumentləri və qaytarma tipi kimi.** | |  |  |
|  |  | **Array of Objects.** | | **Obyektlərin massivi.** | |  |  |
|  |  | **Pointer to Objects and Member Access.** | | **Obyektə və onun elementlərinə əlyetərlik göstəriciləri.** | |  |  |
|  |  | **Dynamic Memory Allocation for Objects and Object Array.** | | **Obyektlər və massiv obyektləri üçün yaddaşın dinamik şəkildə ayrılması.** | |  |  |
|  |  | **this Pointer.** | | **“this” Göstəricisi.** | |  |  |
|  |  | **static Data Member and static Function.** | | **“static” Üzv Verilənləri və “static” funksiyası.** | |  |  |
|  |  | **Constant Member Functions and Constant Objects.** | | **Sabit Üzv Funksiyaları və Sabit obyektlər.** | |  |  |
|  |  | **Friend Function and Friend Classes.** | | **Dost Funksiyalar və Dost Siniflər.** | |  |  |
| **IV BÖLMƏ. İdarəetmə operatorları (Control Statements).** | | | | | | | |
|  |  | **Introduction Part 1: Algorithms. Pseudocode. Control Structures.** | |  | |  |  |
|  |  | **if Selection Statement. if...else Double-Selection Statement.** | |  | |  |  |
|  |  | **while Repetition Statement.** | |  | |  |  |
|  |  | **Formulating Algorithms: Counter-Controlled Repetition. Sentinel-Controlled Repetition. Nested Control Statements.** | |  | |  |  |
|  |  | **Assignment Operators. Increment and Decrement Operators.** | |  | |  |  |
|  |  | **(Optional) Software Engineering Case Study: Identifying Class Attributes in the ATM System.** | |  | |  |  |
|  |  | **Introduction Part 2: Essentials of Counter-Controlled Repetition.** | |  | |  |  |
|  |  | **for Repetition Statement. Examples Using the for Statement.** | |  | |  |  |
|  |  | **do...while Repetition Statement.** | |  | |  |  |
|  |  | **switch Multiple-Selection Statement.** | |  | |  |  |
|  |  | **break and continue Statements.** | |  | |  |  |
|  |  | **Logical Operators.** | |  | |  |  |
|  |  | **Confusing Equality (==) and Assignment (=) Operators.** | |  | |  |  |
|  |  | **Structured Programming Summary.** | |  | |  |  |
|  |  | **(Optional) Software Engineering Case Study: Identifying Objects' States and Activities in the ATM System.** | |  | |  |  |
| **V BÖLMƏ. Funksiyalar və Rekursiyalara Giriş (Functions and an Introduction to Recursion).** | | | | | | | |
|  |  | **Program Components in C++.** | |  | |  |  |
|  |  | **Math Library Functions.** | |  | |  |  |
|  |  | **Function Definitions with Multiple Parameters.** | |  | |  |  |
|  |  | **Function Prototypes and Argument Coercion.** | |  | |  |  |
|  |  | **C++ Standard Library Header Files.** | |  | |  |  |
|  |  | **Case Study: Random Number Generation.** | |  | |  |  |
|  |  | **Case Study: Game of Chance and Introducing enum.** | |  | |  |  |
|  |  | **Storage Classes.** | |  | |  |  |
|  |  | **Scope Rules.** | |  | |  |  |
|  |  | **Function Call Stack and Activation Records.** | |  | |  |  |
|  |  | **Functions with Empty Parameter Lists.** | |  | |  |  |
|  |  | **Inline Functions.** | |  | |  |  |
|  |  | **References and Reference Parameters.** | |  | |  |  |
|  |  | **Default Arguments.** | |  | |  |  |
|  |  | **Unary Scope Resolution Operator.** | |  | |  |  |
|  |  | **Function Overloading.** | |  | |  |  |
|  |  | **Function Templates.** | |  | |  |  |
|  |  | **Recursion.** | |  | |  |  |
|  |  | **Example Using Recursion: Fibonacci Series.** | |  | |  |  |
|  |  | **Recursion vs. Iteration.** | |  | |  |  |
|  |  | **(Optional) Software Engineering Case Study: Identifying Class Operations in the ATM System.** | |  | |  |  |
| **VI BÖLMƏ. Massivlər və Vektorlar (Arrays and Vectors).** | | | | | | | |
|  |  | **Arrays. Declaring Arrays.** | |  | |  |  |
|  |  | **Examples Using Arrays.** | |  | |  |  |
|  |  | **Passing Arrays to Functions.** | |  | |  |  |
|  |  | **Case Study: Class GradeBook Using an Array to Store Grades.** | |  | |  |  |
|  |  | **Searching Arrays with Linear Search.** | |  | |  |  |
|  |  | **Sorting Arrays with Insertion Sort.** | |  | |  |  |
|  |  | **Multidimensional Arrays.** | |  | |  |  |
|  |  | **Case Study: Class GradeBook Using a Two-Dimensional Array.** | |  | |  |  |
|  |  | **Introduction to C++ Standard Library Class Template vector.** | |  | |  |  |
|  |  | **(Optional) Software Engineering Case Study: Collaboration Among Objects in the ATM System.** | |  | |  |  |
| **VII BÖLMƏ. Göstəricilər və Göstərici-Sətirləri (Pointers and Pointer-Based Strings).** | | | | | | | |
|  |  | **Pointer Variable Declarations and Initialization.** | |  | |  |  |
|  |  | **Pointer Operators.** | |  | |  |  |
|  |  | **Passing Arguments to Functions by Reference with Pointers.** | |  | |  |  |
|  |  | **Using const with Pointers.** | |  | |  |  |
|  |  | **Selection Sort Using Pass-by-Reference.** | |  | |  |  |
|  |  | **sizeof Operators.** | |  | |  |  |
|  |  | **Pointer Expressions and Pointer Arithmetic.** | |  | |  |  |
|  |  | **Relationship Between Pointers and Arrays.** | |  | |  |  |
|  |  | **Arrays of Pointers.** | |  | |  |  |
|  |  | **Case Study: Card Shuffling and Dealing Simulation.** | |  | |  |  |
|  |  | **Function Pointers.** | |  | |  |  |
|  |  | **Introduction to Pointer-Based String Processing.** | |  | |  |  |
| **VIII BÖLMƏ. Siniflər: Dərin yanaşma (Classes: A Deeper Look).** | | | | | | | |
|  |  | **Introduction: Part 1** | |  | |  |  |
|  |  | **Time Class Case Study.** | |  | |  |  |
|  |  | **Class Scope and Accessing Class Members.** | |  | |  |  |
|  |  | **Separating Interface from Implementation.** | |  | |  |  |
|  |  | **Access Functions and Utility Functions.** | |  | |  |  |
|  |  | **Time Class Case Study: Constructors with Default Arguments.** | |  | |  |  |
|  |  | **Destructors.** | |  | |  |  |
|  |  | **When Constructors and Destructors Are Called.** | |  | |  |  |
|  |  | **Time Class Case Study: A Subtle TrapReturning a Reference to a private Data Member.** | |  | |  |  |
|  |  | **Default Memberwise Assignment.** | |  | |  |  |
|  |  | **Software Reusability.** | |  | |  |  |
|  |  | **(Optional) Software Engineering Case Study: Starting to Program the Classes of the ATM System.** | |  | |  |  |
|  |  | **Introduction: Part 2** | |  | |  |  |
|  |  | **const (Constant) Objects and const Member Functions.** | |  | |  |  |
|  |  | **Composition: Objects as Members of Classes.** | |  | |  |  |
|  |  | **friend Functions and friend Classes.** | |  | |  |  |
|  |  | **Using the this Pointer.** | |  | |  |  |
|  |  | **Dynamic Memory Management with Operators new and delete.** | |  | |  |  |
|  |  | **static Class Members.** | |  | |  |  |
|  |  | **Data Abstraction and Information Hiding.** | |  | |  |  |
|  |  | **Container Classes and Iterators.** | |  | |  |  |
|  |  | **Proxy Classes.** | |  | |  |  |
| **IX BÖLMƏ. Həddən artıq yükləmə operatoru (Operator Overloading).** | | | | | | | |
|  |  | **Introduction: Operator Overloading.** | |  | |  |  |
|  |  | **Fundamentals of Operator Overloading.** | |  | |  |  |
|  |  | **Restrictions on Operator Overloading.** | |  | |  |  |
|  |  | **Operator Functions as Class Members vs. Global Functions.** | |  | |  |  |
|  |  | **Overloadable Operators.** | | **Həddən artıq yüklənilmiş operatorlar.** | |  |  |
|  |  | **Syntax of Operator Overloading.** | | **Həddən artıq yükləmə operatorunun sintaksisi.** | |  |  |
|  |  | **Rules of Operator Overloading.** | | **Operatorların həddən artıq yüklənilməsi qaydaları.** | |  |  |
|  |  | **Overloading Stream Insertion and Stream Extraction Operators.** | |  | |  |  |
|  |  | **Overloading Unary Operators.** | | **Unar operatorların həddən artıq yüklənilməsi.** | |  |  |
|  |  | **Overloading Binary Operators.** | | **Binar operatorların həddən artıq yüklənilməsi.** | |  |  |
|  |  | **Operator Overloading with Member and Non Member Functions.** | | **Üzv və üzv olmayan funksiyalarla operatorların həddən artıq yüklənilməsi.** | |  |  |
|  |  | **Data Conversion: Basic – User Defined and User Defined – User Defined.** | | **Verilənlərin konvertasiyası: Əsas verilənlərin istifadəçi verilənlərinə və istifadəçi verilənlərinin istifadəçi verilənlərinə konvertasiyası.** | |  |  |
|  |  | **Introduction: String and Array Objects.** | |  | |  |  |
|  |  | **Case Study: Array Class.** | |  | |  |  |
|  |  | **Converting between Types.** | |  | |  |  |
|  |  | **Case Study: String Class.** | |  | |  |  |
|  |  | **Overloading ++ and --.** | |  | |  |  |
|  |  | **Case Study: A Date Class.** | |  | |  |  |
|  |  | **Standard Library Class string.** | |  | |  |  |
|  |  | **Explicit Constructors** | | **Aşkar Konstruktorlar.** | |  |  |
| **X BÖLMƏ. Varislik (Object-Oriented Programming: Inheritance).** | | | | | | | |
|  |  | **Base Classes and Derived Classes.** | | **Baza siniflər və törəmə siniflər.** | |  |  |
|  |  | **protected Members.** | | **“protected” Üzvlər.** | |  |  |
|  |  | **protected Access Specifier.** | | **“protected” əlyetərlik spesifikatoru.** | |  |  |
|  |  | **Derived Class Declaration.** | | **Törəmə sinifinin elan edilməsi.** | |  |  |
|  |  | **Member Function Overriding.** | | **Üzv Funksiyanın yenidən təyin olunması.** | |  |  |
|  |  | **Relationship between Base Classes and Derived Classes.** | | **Baza və törəmə siniflər arasındakı bağlantılar.** | |  |  |
|  |  | **Constructors and Destructors in Derived Classes** | | **Törəmə siniflərdə konstruktorlar və destruktorlar.** | |  |  |
|  |  | **Inheritance. The essence of the principle of inheritance.** | | **Varislik. Varislik prinsipinin mahiyyəti.** | |  |  |
|  |  | **Forms of Inheritance: single, multiple, multilevel, hierarchical, hybrid, multipath.** | | **Varisliyin formaları. Təkvarislik. Çoxvarislik. Çoxsəviyyəli, iyerarxik, hibrid, çoxcığırlı varisliklər.** | |  |  |
|  |  | **Multipath Inheritance and Virtual Base Class.** | | **Çoxcığırlı varislik və virtual baza sinifi.** | |  |  |
|  |  | **Constructor Invocation in Single and Multiple Inheritances.** | | **Tək və çoxsaylı varisdə konstruktorun çağırılması.** | |  |  |
|  |  | **Destructor in Single and Multiple Inheritances.** | | **Tək və çoxsaylı varisdə destruktor.** | |  |  |
|  |  | **public, protected and private Inheritance.** | | **public, protected və private varislikləri.** | |  |  |
|  |  | **Software Engineering with Inheritance.** | | **Varisliyin köməyi ilə proqram təminatının işlənilməsi.** | |  |  |
| **XI BÖLMƏ. Polymorphism and Dynamic Binding (Polimorfizm və Dinamik Bağlanma).** | | | | | | | |
|  |  | **Object-Oriented Programming: Polymorphism.** | |  | |  |  |
|  |  | **Polymorphism Examples.** | |  | |  |  |
|  |  | **Types of Polymorphism.** | | **Polimorfizmin növləri.** | |  |  |
|  |  | **Relationships Among Objects in an Inheritance Hierarchy.** | |  | |  |  |
|  |  | **Type Fields and switch Statements.** | |  | |  |  |
|  |  | **Virtual methods.** | | **Virtual metodlar.** | |  |  |
|  |  | **Constructor. Destruktor.** | | **Konstruktor. Destruktor.** | |  |  |
|  |  | **Dynamic objects. Dynamic methods.** | | **Dinamiki obyektlər. Dinamiki metodlar.** | |  |  |
|  |  | **Schedule virtual and dynamic methods.** | | **Virtual və dinamiki metodlar cədvəli.** | |  |  |
|  |  | **Need of Virtual Function.** | | **Virtual funksiyaya olan tələbatlar.** | |  |  |
|  |  | **Pointer to Derived Class.** | | **Törəmə sinif göstəricisi.** | |  |  |
|  |  | **Definition of Virtual Functions.** | | **Virtual funksiyaların təyinatı.** | |  |  |
|  |  | **Array of Pointers to Base Class.** | | **Baza sinifin göstəriciləri massivi.** | |  |  |
|  |  | **Pure Virtual functions and Abstract Class.** | | **Saf virtual funksiyalar və abstrakt siniflər.** | |  |  |
|  |  | **Case Study: Payroll System Using Polymorphism.** | |  | |  |  |
|  |  | **(Optional) Polymorphism, Virtual Functions and Dynamic Binding "Under the Hood".** | |  | |  |  |
|  |  | **Case Study: Payroll System Using Polymorphism and Run-Time Type Information with Downcasting, dynamic\_cast, typeid and type\_info** | |  | |  |  |
|  |  | **Virtual Destructors.** | | **Virtual Destruktorlar.** | |  |  |
|  |  | **reinterpret\_cast Operator.** | | **reinterpret\_cast Operatoru.** | |  |  |
|  |  | **Run‐Time Type Information.** | | **İcra müddəti tipi barədə informasiya.** | |  |  |
|  |  | **dynamic\_cast Operator.** | | **dynamic\_cast Operatoru.** | |  |  |
|  |  | **typeid Operator.** | | **typeid Operatoru.** | |  |  |
|  |  | **(Optional) Software Engineering Case Study: Incorporating Inheritance into the ATM System.** | |  | |  |  |
| **XII BÖLMƏ. Şablonlar (Templates).** | | | | | | | |
|  |  | **Function Templates.** | |  | |  |  |
|  |  | **Overloading Function Templates.** | |  | |  |  |
|  |  | **Class Templates.** | |  | |  |  |
|  |  | **Nontype Parameters and Default Types for Class Templates.** | |  | |  |  |
|  |  | **Notes on Templates and Inheritance.** | |  | |  |  |
|  |  | **Notes on Templates and Friends.** | |  | |  |  |
|  |  | **Notes on Templates and static Members.** | |  | |  |  |
| **XIII BÖLMƏ. Konsol və Fayl Girişi / Çıxışı üşün Axın hesablaması**  **(Stream Computation for Console and File Input / Output).** | | | | | | | |
|  |  | **Streams. Stream Input. Stream Output.** | |  | |  |  |
|  |  | **Unformatted I/O using read, write and gcount.** | |  | |  |  |
|  |  | **Stream Class Hierarchy for Console Input / Output.** | | **Konsol Girişi / Çıxışı üşün axınların sinifləri iyerarxiyası.** | |  |  |
|  |  | **Testing Stream Errors**  **Unformatted Input / Output.** | | **Formatlanmamış Girişi / Çıxışı üşün axınların səhvlərinin testləşdirilməsi.** | |  |  |
|  |  | **Formatted Input / Output with ios Member functions and Flags.** | | **ios-un üzv-funksiyaları və bayraqları ilə Formatlanmış Girişi / Çıxışı.** | |  |  |
|  |  | **Introduction to Stream Manipulators.** | |  | |  |  |
|  |  | **Formatting with Manipulators.** | | **Manipulyatorların köməyi ilə formatlanma.** | |  |  |
|  |  | **Stream Format States and Stream Manipulators.** | |  | |  |  |
|  |  | **Stream Operator Overloading.** | | **Operatorların həddən artıq yüklənilməsi axını.** | |  |  |
|  |  | **File Input/output with Streams.** | | **Axınların köməyi ilə Faylların Girişi / Çıxışı.** | |  |  |
|  |  | **File Stream Class Hierarchy.** | | **Faylların axını siniflərinin iyerarxiyası.** | |  |  |
|  |  | **Opening and Closing Files.** | | **Faylların açılması və bağlanması.** | |  |  |
|  |  | **Read/Write from File.** | | **Faylların oxunulması və yazılması.** | |  |  |
|  |  | **File Access Pointers and their.** | | **Fayllara və onların manipulyatorlarına əlyetərliyin göstəriciləri.** | |  |  |
|  |  | **Sequential and Random Access to File.** | | **Fayllara ardıcıl və təsadüfi əlyetərlik.** | |  |  |
|  |  | **Stream Error States.** | |  | |  |  |
|  |  | **Testing Errors during File Operations.** | | **Fayllarla iş zamanı səhvlərin testləşdirilməsi.** | |  |  |
|  |  | **Tying an Output Stream to an Input Stream** | |  | |  |  |
| **XIV BÖLMƏ. İstisnaların işlənilməsi (Exception Handling).** | | | | | | | |
|  |  | **Exception-Handling Overview.** | |  | |  |  |
|  |  | **Example: Handling an Attempt to Divide by Zero.** | |  | |  |  |
|  |  | **When to Use Exception Handling.** | |  | |  |  |
|  |  | **Rethrowing an Exception.** | |  | |  |  |
|  |  | **Exception Specifications.** | |  | |  |  |
|  |  | **Processing Unexpected Exceptions.** | |  | |  |  |
|  |  | **Stack Unwinding.** | |  | |  |  |
|  |  | **Constructors, Destructors and Exception Handling.** | |  | |  |  |
|  |  | **Exceptions and Inheritance.** | |  | |  |  |
|  |  | **Processing new Failures.** | |  | |  |  |
|  |  | **Class auto\_ptr and Dynamic Memory Allocation.** | |  | |  |  |
|  |  | **Standard Library Exception Hierarchy.** | |  | |  |  |
|  |  | **Other Error-Handling Techniques.** | |  | |  |  |
| **XV BÖLMƏ. Proseslər (Processes).** | | | | | | | |
|  |  | **Introduction: File Processing.** | |  | |  |  |
|  |  | **The Data Hierarchy.** | |  | |  |  |
|  |  | **Files and Streams.** | |  | |  |  |
|  |  | **Creating a Sequential File.** | |  | |  |  |
|  |  | **Reading Data from a Sequential File.** | |  | |  |  |
|  |  | **Updating Sequential Files.** | |  | |  |  |
|  |  | **Random-Access Files.** | |  | |  |  |
|  |  | **Creating a Random-Access File.** | |  | |  |  |
|  |  | **Writing Data Randomly to a Random-Access File.** | |  | |  |  |
|  |  | **Reading from a Random-Access File Sequentially.** | |  | |  |  |
|  |  | **Case Study: A Transaction-Processing Program.** | |  | |  |  |
|  |  | **Input/Output of Objects.** | |  | |  |  |
|  |  | **Introduction: Class string and String Stream Processing.** | |  | |  |  |
|  |  | **string Assignment and Concatenation.** | |  | |  |  |
|  |  | **Comparing strings.** | |  | |  |  |
|  |  | **Substrings.** | |  | |  |  |
|  |  | **Swapping strings.** | |  | |  |  |
|  |  | **string Characteristics.** | |  | |  |  |
|  |  | **Finding Strings and Characters in a string.** | |  | |  |  |
|  |  | **Replacing Characters in a string.** | |  | |  |  |
|  |  | **Inserting Characters into a string.** | |  | |  |  |
|  |  | **Conversion to C-Style Pointer-Based char \* Strings.** | |  | |  |  |
|  |  | **Iterators.** | |  | |  |  |
|  |  | **String Stream Processing.** | |  | |  |  |
| **CƏMİ:** | | | | | | **75** | **30M/45L** |
| **ƏDƏBİYYAT SİYAHISI** | | | | | |  |  |
| **1. References :**   1. **Robert Lafore, “Object Oriented Programming in C++”, 4th Edition 2002, Sams Publication** 2. **Daya Sagar Baral and Diwakar Baral, “The Secrets of Object Oriented Programming in C++”, 1st Edition 2010, Bhundipuran Prakasan** 3. **Harvey M. Deitel and Paul J. Deitel, “C++ How to Program”, 3rd Edition 2001, Pearson Education Inc.** 4. **Harvey M. Deitel and Paul J. Deitel, “C++ How to Program”, Fifth Edition** 5. **D. S. Malik, “C++ Programming”, 3rd Edition 2007, Thomson Course Technology** 6. **Herbert Schildt, “C++: The Complete Reference”, 4th Edition 2003, Tata McGraw Hill** 7. [**http://www.ioenotes.edu.np/ioe-syllabus/object-oriented-programming-oop-cpp-439**](http://www.ioenotes.edu.np/ioe-syllabus/object-oriented-programming-oop-cpp-439) 8. [**https://docs.oracle.com/javase/tutorial/java/concepts/index.html**](https://docs.oracle.com/javase/tutorial/java/concepts/index.html) 9. [**http://znc.es/Prentice Hall C++ How to Program 5th Edition.pdf**](http://znc.es/Prentice%20Hall%20C++%20How%20to%20Program%205th%20Edition.pdf) 10. [**http://www.ioenotes.edu.np/ioe-bct-syllabus**](http://www.ioenotes.edu.np/ioe-bct-syllabus) 11. [**https://codescracker.com/**](https://codescracker.com/) 12. [**http://www.deitel.com/books/cpphtp5/index.html**](http://www.deitel.com/books/cpphtp5/index.html)   **2. Following web sites may be referred:**   1. [**http://www.osborne.com**](http://www.osborne.com)   **3. The required material for practical can be downloaded from the site:**   1. [**http://www.sun.java.com**](http://www.sun.java.com) | | | | | |  |  |