



यशवंतराव चव्हाण शिक्षण प्रसारक मंडळाचे

**दादासाहेब दिगंबर शंकर पाटील**

**कला, वाणिज्य व विज्ञान महाविद्यालय, एरंडोल, जि. जळगाव**

YASHWANTRAO CHAVAN SHIKSHAN PRASARAK MANDAL'S

**DADASAHEB DIGAMBAR SHANKAR PATIL**

**ARTS, COMMERCE & SCIENCE COLLEGE, ERANDOL DIST. JALGAON**

Website-www.ddsp.ac.in

E-Mail –ddspcera@rediffmail.com

ISO 9001 : 2015

Dr. A. J. Patil (M. E., Ph. D.)

Principal

**DEPARTMENT OF COMPUTER SCIENCE**

**UG (COURSE OUTCOMES)**

CLASS	COURSE	OUTCOMES (Students will be able to)
F.Y.B.Sc. (Computer Science) : Semester: I	Paper: I: CS111: Essentials of Computer	<ul style="list-style-type: none"><li>✓ Students will be able to understand what is Computer and Basic concepts of computer.</li><li>✓ Student will aware about various types of Computers, types of input and output devices.</li><li>✓ Student will be able to preparation of Algorithm and Flowchart of Program.</li><li>✓ Student will learn computer networks, its types and basics of Internet and describe ways computer networks are used to communicate and share resources.</li><li>✓ Student will understand computer viruses and its types.</li></ul>
	Paper: II : CS112 C programming Language I	<ul style="list-style-type: none"><li>✓ To make the student learn a programming language.</li><li>✓ To learn problem solving techniques.</li><li>✓ To teach the student to write programs in C and to solve the problems.</li></ul>
	Paper: III: CS113: Lab On Essentials Of Computer and C programming	<ul style="list-style-type: none"><li>✓ Student will acquire knowledge about different types of computer hardware, software and browsers.</li><li>✓ Student will be able to create e-mail account and sending e-mails with attachment.</li><li>✓ Student will be able to install various devices as well as operating system in the computer.</li><li>✓ Describe various types of networks network standards and communication software.</li><li>✓ Read, understand and trace the execution of programs written in C language.</li><li>✓ Write the C code for a given algorithm.</li><li>✓ Implement Programs with pointers and arrays, perform pointer arithmetic, and use the pre-processor.</li><li>✓ Write programs that perform operations using derived data types.</li></ul>

<b>Semester : II</b>	<b>Paper: I : CS121 Internet Computing</b>	<ul style="list-style-type: none"> <li>✓ Students will be able to Design a basic web site using HTML5 and CSS3 to demonstrate responsive web design.</li> <li>✓ Know the different page types on websites and it's navigations.</li> <li>✓ Student will be able to create web pages using different text styles.</li> <li>✓ Student will be able to design website using HTML language.</li> </ul>
	<b>Paper: II : CS122 C Programming Language II</b>	<ul style="list-style-type: none"> <li>✓ Create algorithms to solve simple programming problems.</li> <li>✓ Describe and employ strategies that are useful in debugging.</li> <li>✓ Design, implement, test and debug programs that use calculations and selections.</li> <li>✓ Design, implement, test and debug programs that use loops and arrays.</li> <li>✓ Design, implement, test and debug programs that use functions.</li> <li>✓ Design, implement, test and debug programs that use arrays for character strings and that use pointers for character strings.</li> </ul>
	<b>Paper: III : CS123Lab course on Internet Computing and C Programming Language II</b>	<ul style="list-style-type: none"> <li>✓ Students will be able to Design a basic web site using HTML5 and CSS3 to demonstrate responsive web design.</li> <li>✓ Know the different page types on websites and it's navigations.</li> <li>✓ Student will be able to create web pages using different text styles.</li> <li>✓ Student will be able to design website using HTML language.</li> <li>✓ Design, implement, test and debug programs that use loops and arrays.</li> <li>✓ Design, implement, test and debug programs that use functions.</li> <li>✓ Design, implement, test and debug programs that use arrays for character strings and that use pointers for character strings.</li> </ul>
<b>S.Y.B.Sc. (Computer Science) : Semester: III</b>	<b>Paper I :CS- DSC 2 C COMP 211 Data Structure I</b>	<ul style="list-style-type: none"> <li>✓ To introduce the fundamental concept of data structures and to emphasize the importance of data structures in developing and implementing efficient algorithms.</li> <li>✓ In addition, another objective of the course is to develop effective software engineering practice, emphasizing such principles as decomposition, procedural abstraction, and software reuse.</li> <li>✓ Ability to analyze algorithms and a algorithm correctness.</li> <li>✓ Ability to summarize searching and sorting techniques</li> <li>✓ Ability to describe stack, queue and linked list operation.</li> <li>✓ Ability to have knowledge of tree and graphs concepts.</li> </ul>
	<b>Paper II :CS- DSC C COMP 212 Programming in C++ - I</b>	<ul style="list-style-type: none"> <li>✓ Identify importance of object oriented programming and difference between structured oriented and object oriented programming features.</li> <li>✓ Able to make use of objects and classes for developing programs.</li> <li>✓ Able to use various object oriented concepts to solve different problems</li> <li>✓ To understand how C++ improves C with object-oriented features.</li> </ul>

		<ul style="list-style-type: none"> <li>✓ To learn how to write inline functions for efficiency and performance.</li> <li>✓ To learn the syntax and semantics of the C++ programming language.</li> <li>✓ To learn how to design C++ classes for code reuse.</li> </ul>
	<b>Paper III : COMP 213 Lab on Data Structure I and Programming in C++ - I</b>	<ul style="list-style-type: none"> <li>✓ For a given Search problem (Linear Search and Binary Search) student will able to implement it.</li> <li>✓ For a given problem of Stacks, Queues and linked list student will able to implement it and analyze the same to determine the time and computation complexity.</li> <li>✓ Describe the procedural and object oriented paradigm with concepts of streams, classes, functions, data and objects.</li> <li>✓ Understand dynamic memory management techniques using pointers, constructors, destructors, etc</li> <li>✓ Describe the concept of function overloading, operator overloading, virtual functions and polymorphism.</li> <li>✓ Classify inheritance with the understanding of early and late binding, usage of exception handling, generic programming.</li> <li>✓ Demonstrate the use of various OOPs concepts with the help of programs</li> <li>✓ Students will able to select appropriate data structures as applied to specified problem definition.</li> <li>✓ Students will be able to implement linear and Non-Linear data structures.</li> <li>✓ Implement appropriate sorting/searching technique for given problem.</li> <li>✓ Design advance data structure using Non- Linear data structure.</li> <li>✓ Determine and analyze the complexity of given Algorithms.</li> </ul>
	<b>Paper IV Software &amp; Hardware Installation Skills</b>	<ul style="list-style-type: none"> <li>✓ This course helps student step by step through the typical hardware and operating system problems encountered by technicians, teaching troubleshooting techniques to decipher any problem, and giving you the skills you need to solve them.</li> <li>✓ Students will understand basic concept &amp; structure of computer hardware &amp; networking and apply their knowledge about computer peripherals to identify / rectify problems onboard.</li> <li>✓ able to identify the reasons which require a hardware/software installation/upgrade/replacement</li> <li>✓ able to identify the risks involved when performing an installation/upgrade/replacement and take the necessary precautions to avoid such risks on both human and electronic component terms.</li> <li>✓ discuss precautions which must be taken to avoid risks when replacing/installing/upgrading hardware components.</li> <li>✓ able to install/upgrade/replace internal hardware components and configure the necessary software.</li> </ul>

<b>Semester: IV :</b>	<b>Paper I CS-DSC 2 D COMP 221 Data Structure- II</b>	<ul style="list-style-type: none"> <li>✓ Understand the concept of Dynamic memory management, data types, algorithms, Big O notation.</li> <li>✓ Understand basic data structures such as arrays, linked lists, stacks and queues.</li> <li>✓ Describe the hash function and concepts of collision and its resolution methods</li> <li>✓ Solve problem involving graphs, trees and heaps</li> <li>✓ Apply Algorithm for solving problems like sorting, searching, insertion and deletion of data</li> </ul>
	<b>Paper II CS-DSC 2 D COMP 222 Programming in C++-II</b>	<ul style="list-style-type: none"> <li>✓ Gain a foundation for writing efficient, safe C++ code</li> <li>✓ Learn how to use STL libraries</li> <li>✓ Understand memory pitfalls in C++</li> <li>✓ Know how to expand the C++ memory model</li> <li>✓ Utilize object-oriented for analysis and design</li> <li>✓ Apply multiple inheritance to an application</li> <li>✓ Understand how streams work</li> <li>✓ See how to persist objects via streams</li> <li>✓ Gain experience following C++ guidelines</li> <li>✓ Identify potential bugs in code before they occur</li> </ul>
	<b>Paper III : COMP 223 Lab on Data Structure II and Programming in C++ - II</b>	<ul style="list-style-type: none"> <li>✓ Analyse the efficiency of the designed algorithms.</li> <li>✓ Implement the main data structures: Lists, stacks, queues, trees, graphs and hash tables; and use them to solve computational problems.</li> <li>✓ Use recursive design.</li> <li>✓ Analyse and use the main search, sorting and traversal algorithms in data structures.</li> <li>✓ Design and implement data structures to efficiently solve problems.</li> <li>✓ Remember the characteristics of Procedure and Object Oriented Programming Languages</li> <li>✓ Understand the fundamentals of C++ programming structure, function overloading and constructors.</li> <li>✓ To be able to program using C++ features such as composition of objects, Operator overloading, inheritance, Polymorphism etc.</li> <li>✓ Apply the concepts in object oriented programming in terms of software reuse and managing complexity to solve real-world problems.</li> </ul>
	<b>Paper IV Network Security</b>	<ul style="list-style-type: none"> <li>✓ identify some of the factors driving the need for network security</li> <li>✓ identify and classify particular examples of attacks</li> <li>✓ define the terms vulnerability, threat and attack</li> <li>✓ identify physical points of vulnerability in simple networks</li> <li>✓ compare and contrast symmetric and asymmetric encryption systems and their vulnerability to attack, and explain the characteristics of hybrid systems</li> <li>✓ To understand basics of Cryptography and Network Security.</li> </ul>

		<ul style="list-style-type: none"> <li>✓ To be able to secure a message over insecure channel by various means.</li> <li>✓ To learn about how to maintain the Confidentiality, Integrity and Availability of a data.</li> <li>✓ To understand various protocols for network security to protect against the threats in the networks.</li> </ul>
<b>T .Y. B.Sc. (Computer Science) Semester: V</b>	<b>Paper:I CS 501 System Programming</b>	<ul style="list-style-type: none"> <li>✓ Explain basic concepts in systems programming.</li> <li>✓ Describe UNIX file systems and process control.</li> <li>✓ Utilize UNIX system services in application development.</li> <li>✓ Design and implement system utility programs.</li> <li>✓ Student will learn program development tools, basic testing, timing, profiling and benchmarking, characteristics of physical devices, memory management, device drivers, pseudo-devices, file structures, file I/O (both buffered and unbuffered), processes, shells, inter-process communications, signals, exceptions, pipes, sockets, shared memory and file and record locking.</li> </ul>
	<b>Paper:II CS 502 Database Management System</b>	<ul style="list-style-type: none"> <li>✓ Understand database concepts and structures and query language</li> <li>✓ Understand the E R model and relational model</li> <li>✓ To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.</li> <li>✓ Understand Functional Dependency and Functional Decomposition.</li> <li>✓ Apply various Normalization techniques</li> <li>✓ Perform PL/SQL programming using concept of Cursor Management, Error Handling, Package and Triggers</li> <li>✓ Execute various advance SQL queries related to Transaction Processing &amp; Locking using concept of Concurrency control.</li> <li>✓ Understand query processing and techniques involved in query optimization.</li> <li>✓ Understand the principles of storage structure and recovery management.</li> </ul>
	<b>Paper:III CS 503 Software Engineering</b>	<ul style="list-style-type: none"> <li>✓ An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics</li> <li>✓ An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors</li> <li>✓ An ability to communicate effectively with a range of audiences</li> <li>✓ An ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts</li> </ul>

		<ul style="list-style-type: none"> <li>✓ An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives</li> <li>✓ An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions</li> <li>✓ An ability to acquire and apply new knowledge as needed, using appropriate learning strategies</li> </ul>
	<b>Paper: IV CS 504 Computer Aided Graphics</b>	<ul style="list-style-type: none"> <li>✓ In modern industry, the computer supports all the design and manufacturing functions.</li> <li>✓ Analyze technical drawings using both CAD and basic manual tools.</li> <li>✓ Create mechanical parts for different applications.</li> <li>✓ Apply the stages of the design process from scratch using engineering graphics techniques such as sectional projections, dimensioning and computer-generated drawings (2D).</li> <li>✓ Computer graphics is a fast growing field of study, which has many variable course offerings to accommodate the ever-changing technology.</li> <li>✓ Apply principles of technical drawings to create different 3D models.</li> <li>✓ Utilize SolidWorks surfacing features and methods to create complex solid geometry.</li> <li>✓ Apply ANSI drafting standards to different type of projects.</li> <li>✓ Demonstrate problem-solving skills to master the creation of drawings and assemblies. set the stage for future accreditation of a computer graphics program.</li> </ul>
	<b>Paper: V CS 505 Python Programming – I</b>	<ul style="list-style-type: none"> <li>✓ Students start with tutorials on Python and IDLE (software environment for Python), with embedded exercises.</li> <li>✓ After the first few days, work alternates in blocks of two hours between programming assignments and exercises (and tutorials).</li> <li>✓ While one or more groups continue tutorials and exercises (later only exercises), the other group(s) do programming assignments.</li> <li>✓ In the next block, the groups switch activities: the group(s) that started at assignments, work on tutorials and exercises again, and vice versa.</li> <li>✓ Implement a given algorithm as a computer program (in Python)</li> <li>✓ Adapt and combine standard algorithms to solve a given problem (includes numerical as well as non-numerical algorithms)</li> <li>✓ Adequately use standard programming constructs: repetition, selection, functions, composition, modules, aggregated data (arrays, lists, etc.)</li> </ul>

		<ul style="list-style-type: none"> <li>✓ explain what a given program (in Python) does Identify and repair coding errors in a program</li> <li>✓ Understand and use object based software concepts (constructing OO software will be dealt with in the course Software Engineering)</li> <li>✓ Use library software for (e.g.) building a graphical user interface, web application, or mathematical software.</li> </ul>
	<b>Paper: VI CS 506 (B) Java Programming –I</b>	<ul style="list-style-type: none"> <li>✓ Students will complete software projects comprised of an object-oriented design, implementation, and test plan.</li> <li>✓ Designs will demonstrate the use of good object-oriented design principles including encapsulation and information hiding.</li> <li>✓ The implementation will demonstrate the use of a variety of basic control structures including selection and repetition; classes and objects in a tiered architecture (user interface, controller, and application logic layers); primitive and reference data types including composition; basic AWT components; file-based I/O; and one-dimensional arrays.</li> <li>✓ Use an integrated development environment to write, compile, run, and test simple object-oriented Java programs.</li> <li>✓ Read and make elementary modifications to Java programs that solve real-world problems.</li> <li>✓ Validate input in a Java program.</li> <li>✓ Identify and fix defects and common security issues in code.</li> <li>✓ Document a Java program using Javadoc.</li> <li>✓ Use a version control system to track source code in a project.</li> </ul>
	<b>CS – Lab -507 Lab on Python Programming - I</b>	<ul style="list-style-type: none"> <li>✓ Python programming is intended for software engineers, system analysts, program managers and user support personnel who wish to learn the Python programming language.</li> <li>✓ To learn how to design and program Python applications.</li> <li>✓ To develop problem solving skills and their implementation through Python.</li> <li>✓ Master the fundamentals of writing Python scripts</li> <li>✓ Assignments and class exercises focus on learning how to program. Together they cover the learning outcomes.</li> <li>✓ Exercises proper (during computer lessons) focus more on elements of the Python language and particular libraries. Often, those exercises prepare for using a technique in the assignments.</li> <li>✓ Many exercises are strongly linked with the book.</li> <li>✓ Students are encouraged to study the book along with these exercises.</li> <li>✓ One slot of two hours per week is especially dedicated to studying the book.</li> </ul>

	<b>CS – Lab -508 Computer Aided Graphics</b>	<ul style="list-style-type: none"> <li>✓ Overall, computer technology related programs in the field of industrial technology represent a rapidly emerging area of study</li> <li>✓ It helps the students to implement various graphics techniques</li> <li>✓ It helps the students to implement various graphics algorithms</li> <li>✓ Differentiate between interactive and non-interactive graphics.</li> <li>✓ Study line Drawing and Circle Drawing techniques and algorithms.</li> <li>✓ Perform 2D and 3D transformation on different images.</li> <li>✓ Know about detail working of 2D and 3D clipping and windowing.</li> <li>✓ Understand raster graphics and hidden surface elimination.</li> <li>✓ Rapidly emerging programs must be continuously assessed and monitored to make certain that they are academically appropriate, differentiated from other computer-related programs, and are meeting the needs and expectations of key stakeholders</li> </ul>
	<b>CS – Lab -509 Lab on Java Programming - I</b>	<ul style="list-style-type: none"> <li>✓ Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.</li> <li>✓ Understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.</li> <li>✓ Be aware of the important topics and principles of software development in java.</li> <li>✓ Have the ability to write a computer program to solve specified problems in java.</li> <li>✓ Summarize the strengths and weaknesses of Java programming and the basic concepts of object-oriented programming.</li> <li>✓ Identify Java code utilities in applets, Java packages, and classes.</li> <li>✓ Write Java code using advanced Java features.</li> </ul>
<b>Semester VI :</b>	<b>Paper I CS - 601 Operating System</b>	<ul style="list-style-type: none"> <li>✓ A successful student will be able to understand the basic components of a computer operating system, and the interactions among the various components. The course will cover an introduction on the policies for scheduling, deadlocks, memory management, synchronization, system calls, and file systems. The students will implement solutions via C/C++ programs, and through NACHOS.</li> <li>✓ Describe and explain the fundamental components of a computer operating system.</li> <li>✓ analysis, modeling, and problem solving. Students have the logical, algorithmic, and mathematical capability to model and analyze real-world problems in different application domains, to devise the problem-solving schemes accordingly, and to validate the correctness and effectiveness of the schemes.</li> <li>✓ Identify and define key terms related to operating systems.</li> </ul>



		<ul style="list-style-type: none"> <li>✓ Explain basic concepts related to concurrency and control of concurrent programs.</li> <li>✓ Analyze moderately complex regular expressions.</li> <li>✓ Write simple concurrent programs that are free of problems.</li> </ul>
	<b>Paper II: CS 602 RDBMS</b>	<ul style="list-style-type: none"> <li>✓ Be familiar with the basic issues of transaction processing and concurrency control</li> <li>✓ To teach data security and its importance</li> <li>✓ Analyze the data and data organization needs of organizations;</li> <li>✓ Apply the Entity-Relationship (E-R) Model for building information systems' data models;</li> <li>✓ Transform an E-R diagram into a relational model, and use normalization to create a database relational schema;</li> <li>✓ Discuss the physical database design process of producing an efficient and tuned database;</li> <li>✓ Explain when denormalization is preferred over normalization, and use vertical and horizontal partitioning for data distribution;</li> <li>✓ We will discuss in this subject the evolution of file processing systems, and introduces the role of databases in organizations. It also introduces the core components of a database environment.</li> <li>✓ It explains the relational model, normalization, and how to transform an entity-relationship data diagram into a relational model.</li> </ul>
	<b>Paper III:CS - 603 Computer Network</b>	<ul style="list-style-type: none"> <li>✓ This subject helps to understand</li> <li>✓ Demonstrate expertise in configuring host and network level technical security controls, to include host firewalls, user access controls, host logging, network filtering, intrusion detection, and prevention and encryption at all levels</li> <li>✓ How network works? &amp; types of networks &amp; its applications.</li> <li>✓ It helps to understand the various models.</li> <li>✓ It helps to understand various layers &amp; their functionality.</li> <li>✓ Explain key networking protocols, and their hierarchical relationship in the context of a conceptual model, such as the OSI and TCP/IP framework; be able to articulate the low level data communications and subsequent abstractions that allow networked hosts and applications to communicate across the internet;</li> <li>✓ Students understand the information exchange done across the network with the help of OSI &amp; TCP/IP models.</li> <li>✓ Student understands how errors are captured &amp; handled in network.</li> <li>✓ Student understands various attack &amp; its prevention techniques.</li> </ul>
	<b>Paper IV: CS 604 Theoretical</b>	<ul style="list-style-type: none"> <li>✓ Upon successful completion, students will have the knowledge and skills to:</li> </ul>

	<b>Computer Science</b>	<ul style="list-style-type: none"> <li>✓ Describe and explain key concepts of algorithm design, formal models of computation and complexity theory.</li> <li>✓ Expertly apply techniques of logic, information theory and finite mathematics to the analysis of algorithms.</li> <li>✓ Think critically and assimilate information from different disciplines in order to solve problems related to computer science.</li> <li>✓ Evaluate and critically analyse scientific literature to extend their knowledge in the field.</li> </ul> <p>Communicate and present their knowledge of computer science clearly to diverse audiences.</p>
	<b>Paper V : CS 605 Python Programming – II</b>	<ul style="list-style-type: none"> <li>✓ implement a given algorithm as a computer program (in Python)</li> <li>✓ Adapt and combine standard algorithms to solve a given problem (includes numerical as well as non-numerical algorithms)</li> <li>✓ Adequately use standard programming constructs: repetition, selection, functions, composition, modules, aggregated data (arrays, lists, etc.)</li> <li>✓ explain what a given program (in Python) does identify and repair coding errors in a program</li> <li>✓ understand and use object based software concepts (constructing OO software will be dealt with in the course Software Engineering)</li> <li>✓ use library software for (e.g.) building a graphical user interface, web application, or mathematical software</li> </ul>
	<b>Paper VI: CS 606 (B) JAVA Programming II</b>	<ul style="list-style-type: none"> <li>✓ knowledge of the structure and model of the Java programming language, (knowledge)</li> <li>✓ use the Java programming language for various programming technologies (understanding)</li> <li>✓ develop software in the Java programming language,(application)</li> <li>✓ evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements (analysis)</li> <li>✓ propose the use of certain technologies by implementing them in the Java programming language to solve the given problem(synthesis)</li> <li>✓ choose an engineering approach to solving problems, starting from the acquired knowledge of programming and knowledge of operating systems. (evaluation)</li> </ul>
	<b>Paper VII: CS 607 LAB on Python Programming – II</b>	<ul style="list-style-type: none"> <li>✓ implement a given algorithm as a computer program (in Python)</li> <li>✓ Adapt and combine standard algorithms to solve a given problem (includes numerical as well as non-numerical algorithms)</li> </ul>

		<ul style="list-style-type: none"> <li>✓ Adequately use standard programming constructs: repetition, selection, functions, composition, modules, aggregated data (arrays, lists, etc.)</li> <li>✓ explain what a given program (in Python) does identify and repair coding errors in a program</li> <li>✓ understand and use object based software concepts (constructing OO software will be dealt with in the course Software Engineering)</li> <li>✓ use library software for (e.g.) building a graphical user interface, web application, or mathematical software.</li> </ul>
	<b>Paper VIII: CS LAB 608 Lab on RDBMS</b>	<ul style="list-style-type: none"> <li>✓ On completion of this course, students will be able to :</li> <li>✓ To use SQL &amp; PL/SQL.</li> <li>✓ To perform advanced database operations.</li> <li>✓ Create database tables in postgre SQL.</li> <li>✓ Write and execute simple, nested queries</li> </ul>
	<b>Paper IX : CS 609(B) Lab on JAVA Programming II</b>	<ul style="list-style-type: none"> <li>✓ knowledge of the structure and model of the Java programming language, (knowledge)</li> <li>✓ use the Java programming language for various programming technologies (understanding)</li> <li>✓ develop software in the Java programming language,(application)</li> <li>✓ evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements (analysis)</li> <li>✓ propose the use of certain technologies by implementing them in the Java programming language to solve the given problem(synthesis)</li> <li>✓ choose an engineering approach to solving problems, starting from the acquired knowledge of programming and knowledge of operating systems. (evaluation)</li> </ul>