Zakres wiedzy i umiejętności:

- Zna i posługuje się różnymi sposobami reprezentowania informacji w komputerze: liczb, znaków, obrazów, animacji, dźwięku;
- Opisuje różne systemy operacyjne;
- Zna i przedstawia warstwowy model sieci komputerowych, opisuje ustawienia sieciowe komputera w sieci, zna architekturę klient-serwer, posługuje się terminologią sieciową, potrafi wyznaczyć maski sieci i podsieci na podstawie masek;
- Projektuje relacyjną bazę danych, stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych, potrafi tworzyć raporty na podstawie zapytań do baz danych;
- Znajduje odpowiednie informacje niezbędne do realizacji w środowisku komputerowym projektów z różnych dziedzin;
- Opisuje mechanizmy związane z bezpieczeństwem danych: szyfrowanie, klucz, certyfikat, zapora ogniowa;
- Wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów;
- Zna i stosuje terminologię z zakresu przetwarzania grafiki komputerowej, przetwarza obrazy i filmy w środowisku komputerowym;
- Stosuje podejście algorytmiczne do rozwiązywania problemów z różnych dziedzin, dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji, ocenia zgodność otrzymanego rozwiązania ze specyfikacją problemu, ocenia efektywność otrzymanego rozwiązania;
- Posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów;
- Stosuje rekurencję do rozwiązywania sytuacji problemowych;
- Stosuje podejście zachłanne w rozwiązywaniu problemów;
- Zna i stosuje podstawowe algorytmy w zakresie:
    - Algorytmy operowania na liczbach całkowitych, np.:
        - Reprezentacja liczb w dowolnym systemie pozycyjnym;
        - Sprawdzenie, czy liczba jest liczbą pierwszą;
        - Rozkład liczby na czynniki pierwsze;
        - Rekurencyjna i iteracyjna realizacja algorytmu Euklidesa;
        - Rekurencyjne i iteracyjne obliczanie wartości liczb Fibonacciego;
        - Wydawanie reszty metodą zachłanną;
    - Algorytmy wyszukiwania i porządkowania, np.:
        - Algorytm naiwny i optymalny jednoczesnego wyszukiwania najmniejszego i największego elementu w zbiorze;
        - Algorytmy sortowania ciągu liczb: bąbelkowy, przez wybór, przez wstawianie liniowe lub binarne, przez scalanie, szybki, kubełkowy;
    - Algorytmy numeryczne, np.:
        - Obliczanie wartości pierwiastka kwadratowego;
        - Obliczanie wartości wielomianu za pomocą schematu Hornera;
        - Wyznaczanie miejsc zerowych funkcji metodą połowienia;
        - Całkowanie numeryczne metodą prostokątów, trapezów;
        - Metoda simplex rozwiązywania zadań programowania liniowego;
    - Algorytmy na tekstach, np.:
        - Sprawdzanie, czy ciąg znaków tworzy palindrom, anagram;
        - Porządkowanie alfabetyczne;
        - Wyszukiwanie wzorca w tekście;

        - Obliczanie wartości wyrażenia podanego w postaci odwrotnej notacji polskiej;
    - Algorytmy kompresji i szyfrowania, np.:
        - Kody znaków o zmiennej długości, np. kod Huffmana;
        - Szyfr Cezara;
        - Szyfr przestawiony;
    - Algorytmy badające właściwości geometryczne, np.:
        - Sprawdzanie warunku trójkąta;
        - Badanie położenia punktów względem prostej;
        - Badanie przynależności punktu do odcinka;
        - Przecinanie się odcinków;
        - Przynależność punktu do obszaru;
        - Konstrukcje rekurencyjne: drzewo binarne, dywan Sierpińskiego, płatek Kocha;
- Zna i potrafi tworzyć reprezentację grafową problemu;
- Zna algorytmy operujące na grafach, np.:
    - Przeszukiwanie grafu wszerz (BFS) i w głąb (DFS);
    - SPT (Shortest Processing Time);
    - Dijkstry;
    - Floyda;
    - Wyznaczanie najkrótszej drogi w grafie dla znanej odległości;
    - Kolorowanie grafu;
    - Kruskala;
    - Prima;
- Projektuje rozwiązanie problemu i dobiera odpowiednie struktury danych, w tym struktury dynamiczne;
- Stosuje zasady programowania strukturalnego i modularnego do rozwiązania problemu;
- Oblicza liczbę operacji wykonywanych przez algorytm;
- Szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu;
- Przeprowadza komputerową realizację algorytmu i rozwiązania problemu;
- Sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów;
- Stosuje podstawowe konstrukcje programistyczne w języku C/C++, instrukcje iteracyjne, warunkowe, funkcje, instrukcje wejścia/wyjścia, poprawnie tworzy strukturę programu;
- Dobiera najlepszy algorytm, odpowiednie struktury danych do rozwiązania postawionego problemu;
- Dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązania postawionego problemu;
- Ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania;
- Wyjaśnia źródło błędów w obliczeniach komputerowych: błąd względny, błąd bezwzględny;