

WYMAGANIA EDUKACYJNE DLA ZAWODU TECHNIK PROGRAMISTA

PRZEDMIOT: PRACOWNIA BAZ DANYCH

program nr 351406 Kwalifikacja inf03

Aby uczeń mógł otrzymać ocenę dopuszczającą powinien:

- umieć objaśnić relacyjny model baz danych i rolę systemu zarządzania bazą danych
- umieć wyjaśnić rolę normalizacji bazy danych
- umieć podać zasady projektowania relacyjnych baz danych
- umieć obsługiwać wybrany SZBD
- umieć obsługiwać prostą, relacyjną bazę danych typu desktop za pomocą gotowych tabel i formularzy
- potrafić przeprowadzić w bazie sortowanie danych i jej proste filtrowanie
- umieć obsługiwać gotowe kwerendy
- umieć utworzyć proste formularze i raporty w bazie danych
- potrafić wyjaśnić rolę makr w systemie bazodanowym
- umieć wymienić aspekty bezpieczeństwa baz i danych w bazach
- umieć zalogować się w bazie danych
- umieć obsługiwać wybrany SZBD
- umieć obsługiwać relacyjną bazę danych typu desktop za pomocą gotowych tabel i formularzy
- umieć korzystać z gotowego interfejsu bazy danych
- umieć obsługiwać gotowe kwerendy
- umieć utworzyć proste formularze i raporty w bazie danych
- umieć wybraną metodą powiązać obiekty graficzne z bazą
- umieć tworzyć odsyłacze w bazie danych
- umieć zaimportować proste struktury danych do bazy
- umieć eksportować proste struktury danych z bazy
- umieć wyjaśnić rolę języka SQL w bazach danych
- umieć tworzyć proste zapytania w języku SQL
- umieć wyjaśnić rolę języka VBA w SZBD
- umieć wymienić aspekty bezpieczeństwa baz i danych w bazach
- umieć zalogować się w bazie danych

Aby uczeń mógł otrzymać ocenę dostateczną powinien spełniać wymagania na ocenę dopuszczającą oraz:

- potrafić zaprojektować prostą relacyjną bazę danych
- potrafić tworzyć podstawowe obiekty bazy danych
- potrafić modyfikować z pomocą nauczyciela obiekty bazy danych
- umieć wiązać ze sobą tabele bazy danych relacjami

- umieć modyfikować gotowe formularze i raporty i inne obiekty bazodanowe
- umieć selekcjonować informacje z bazy danych za pomocą samodzielnie konstruowanych filtrów
- umieć uzyskiwać informacje z bazy danych za pomocą samodzielnie konstruowanych w sposób wizualno-obiektowy prostych kwerend (zapytań) wybierających
- umieć samodzielnie modyfikować zawartość bazy danych poprzez konstruowane w sposób wizualno-obiektowy proste zapytania funkcjonalne
- potrafić tworzyć proste makra w systemie bazodanowym
- umieć tworzyć podstawowe obiekty bazy danych
- umieć modyfikować obiekty bazy danych
- umieć modyfikować obiekty SZBD
- umieć importować dane do bazy z różnych źródeł i eksportować dane z bazy w różnych formatach
- umieć tworzyć zapytania w języku SQL
- automatyzować zadania formularzy i raportów za pomocą przycisków akcji
- umieć tworzyć kod VBA za pomocą kreatorów formantów
- umieć tworzyć proste procedury obsługi zdarzeń w języku VBA
- potrafić użyć elementarnych konstrukcji algorytmicznych VBA
- potrafić użyć zmiennej w kodzie VBA
- umieć zastosować proste mechanizmy zabezpieczania baz danych

Aby uczeń mógł otrzymać ocenę dobrą powinien spełniać wymagania na ocenę dostateczną oraz:

- potrafić zaprojektować umiarkowanie złożoną relacyjną bazę danych
- potrafić tworzyć tabele bazy danych z polami odnośników
- umieć samodzielnie modyfikować projekty tabel bazy danych
- potrafić tworzyć umiarkowanie złożone kwerendy wybierające i funkcjonalne w sposób wizualno-obiektowy
- umieć projektować formularze i raporty oraz samodzielnie je modyfikować
- potrafić tworzyć grupy makr i makra warunkowe w systemie bazodanowym
- tworzyć proste zabezpieczenia baz danych
- umieć wykorzystywać mechanizmy zabezpieczeń baz danych
- umieć samodzielnie modyfikować projekty tabel bazy danych
- umieć optymalizować pracę bazy danych
- umieć wykorzystywać różne mechanizmy zabezpieczeń baz danych
- umieć zastosować replikację bazy danych
- umieć tworzyć algorytmy VBA z użyciem typowych konstrukcji algorytmicznych
- umieć tworzyć procedury i funkcje w kodzie VBA
- potrafić obsługiwać zmienne w algorytmach VBA

- tworzyć złożone zapytania w języku SQL

Aby uczeń mógł otrzymać ocenę bardzo dobrą powinien spełniać wymagania na ocenę dobrą oraz:

- umieć projektować złożone relacyjne bazy danych
- umieć zadbać o integralność i spójność danych w bazie
- umieć wykorzystać mechanizmy kontroli wprowadzania danych
- umieć tworzyć złożone kwerendy w bazie danych
- umieć tworzyć złożone formularze z podformularzami
- automatyzować zadania formularzy i raportów za pomocą przycisków akcji
- automatyzować zadania całej bazy danych za pomocą makr
- umieć projektować złożone relacyjne bazy danych
- umieć zadbać o integralność i spójność danych w bazie
- umieć replikować bazy danych
- umieć wykorzystać mechanizmy kontroli wprowadzania danych
- potrafić kontrolować interfejs bazy danych z poziomu kodu VBA
- umieć komunikować się z użytkownikiem bazy z poziomu kodu VBA
- umieć korzystać z mechanizmów ADO i DAO w tworzonych aplikacjach baz danych
- potrafić tworzyć umiarkowanie złożone procedury i funkcje w kodzie VBA
- potrafić wykonywać kod SQL z poziomu kodu VBA

Aby uczeń mógł otrzymać ocenę celującą powinien spełniać wymagania na ocenę bardzo dobrą oraz jedno z poniższych kryteriów:

- osiągnie sukces w konkursie związanym z tematyką zajęć
- wykona dodatkowy projekt, wykazując się dodatkowymi umiejętnościami, zgodny z nauczonym przedmiotem. Projekt ma być ustalony z nauczycielem i realizowany systematycznie, a postępy będą prezentowane nauczycielowi w ciągu całego roku szkolnego lub semestru
- osiągnie średnią ocen zgodną z WSO, klasyfikującą do wystawienia oceny celującej

PRZEDMIOT: PROJEKTOWANIE I ADMINISTROWANIE BAZAMI DANYCH

program nr 351406 Kwalifikacja inf03

Aby uczeń mógł otrzymać ocenę dopuszczającą powinien:

- umieć objaśnić relacyjny model baz danych i rolę systemu zarządzania bazą danych
- umieć wyjaśnić rolę normalizacji bazy danych
- umieć podać zasady projektowania relacyjnych baz danych
- umieć obsługiwać wybrany SZBD
- umieć obsługiwać prostą, relacyjną bazę danych typu desktop za pomocą gotowych tabel i formularzy
- potrafić przeprowadzić w bazie sortowanie danych i jej proste filtrowanie
- umieć obsługiwać gotowe kwerendy
- umieć utworzyć proste formularze i raporty w bazie danych
- potrafić wyjaśnić rolę makr w systemie bazodanowym
- umieć wymienić aspekty bezpieczeństwa baz i danych w bazach
- umieć zalogować się w bazie danych
- umieć obsługiwać wybrany SZBD
- umieć obsługiwać relacyjną bazę danych typu desktop za pomocą gotowych tabel i formularzy
- umieć korzystać z gotowego interfejsu bazy danych
- umieć obsługiwać gotowe kwerendy
- umieć utworzyć proste formularze i raporty w bazie danych
- umieć wybraną metodą powiązać obiekty graficzne z bazą
- umieć tworzyć odsyłacze w bazie danych
- umieć zaimportować proste struktury danych do bazy
- umieć eksportować proste struktury danych z bazy
- umieć wyjaśnić rolę języka SQL w bazach danych
- umieć tworzyć proste zapytania w języku SQL
- umieć wyjaśnić rolę języka VBA w SZBD
- umieć wymienić aspekty bezpieczeństwa baz i danych w bazach
- umieć zalogować się w bazie danych

Aby uczeń mógł otrzymać ocenę dostateczną powinien spełniać wymagania na ocenę dopuszczającą oraz:

- potrafić zaprojektować prostą relacyjną bazę danych
- potrafić tworzyć podstawowe obiekty bazy danych
- potrafić modyfikować z pomocą nauczyciela obiekty bazy danych
- umieć wiązać ze sobą tabele bazy danych relacjami
- umieć modyfikować gotowe formularze i raporty i inne obiekty bazodanowe

- umieć selekcjonować informacje z bazy danych za pomocą samodzielnie konstruowanych filtrów
- umieć uzyskiwać informacje z bazy danych za pomocą samodzielnie konstruowanych w sposób wizualno-objektowy prostych kwerend (zapytań) wybierających
- umieć samodzielnie modyfikować zawartość bazy danych poprzez konstruowane w sposób wizualno-objektowy proste zapytania funkcjonalne
- potrafić tworzyć proste makra w systemie bazodanowym
- umieć tworzyć podstawowe obiekty bazy danych
- umieć modyfikować obiekty bazy danych
- umieć modyfikować obiekty SZBD
- umieć importować dane do bazy z różnych źródeł i eksportować dane z bazy w różnych formatach
- umieć tworzyć zapytania w języku SQL
- automatyzować zadania formularzy i raportów za pomocą przycisków akcji
- umieć tworzyć kod VBA za pomocą kreatorów formantów
- umieć tworzyć proste procedury obsługi zdarzeń w języku VBA
- potrafić użyć elementarnych konstrukcji algorytmicznych VBA
- potrafić użyć zmiennej w kodzie VBA
- umieć zastosować proste mechanizmy zabezpieczenia baz danych

Aby uczeń mógł otrzymać ocenę dobrą powinien spełniać wymagania na ocenę dostateczną oraz:

- potrafić zaprojektować umiarkowanie złożoną relacyjną bazę danych
- potrafić tworzyć tabele bazy danych z polami odnośników
- umieć samodzielnie modyfikować projekty tabel bazy danych
- potrafić tworzyć umiarkowanie złożone kwerendy wybierające i funkcjonalne w sposób wizualno-objektowy
- umieć projektować formularze i raporty oraz samodzielnie je modyfikować
- potrafić tworzyć grupy makr i makra warunkowe w systemie bazodanowym
- tworzyć proste zabezpieczenia baz danych
- umieć wykorzystywać mechanizmy zabezpieczeń baz danych
- umieć samodzielnie modyfikować projekty tabel bazy danych
- umieć optymalizować pracę bazy danych
- umieć wykorzystywać różne mechanizmy zabezpieczeń baz danych
- umieć zastosować replikację bazy danych
- umieć tworzyć algorytmy VBA z użyciem typowych konstrukcji algorytmicznych
- umieć tworzyć procedury i funkcje w kodzie VBA
- potrafić obsługiwać zmienne w algorytmach VBA
- tworzyć złożone zapytania w języku SQL;

Aby uczeń mógł otrzymać ocenę bardzo dobrą powinien spełniać wymagania na ocenę dobrą oraz:

- umieć projektować złożone relacyjne bazy danych
- umieć zadbać o integralność i spójność danych w bazie
- umieć wykorzystać mechanizmy kontroli wprowadzania danych
- umieć tworzyć złożone kwerendy w bazie danych;
- umieć tworzyć złożone formularze z podformularzami
- automatyzować zadania formularzy i raportów za pomocą przycisków akcji
- automatyzować zadania całej bazy danych za pomocą makr
- umieć projektować złożone relacyjne bazy danych
- umieć zadbać o integralność i spójność danych w bazie
- umieć replikować bazy danych
- umieć wykorzystać mechanizmy kontroli wprowadzania danych
- potrafić kontrolować interfejs bazy danych z poziomu kodu VBA
- umieć komunikować się z użytkownikiem bazy z poziomu kodu VBA
- umieć korzystać z mechanizmów ADO i DAO w tworzonych aplikacjach baz danych
- potrafić tworzyć umiarkowanie złożone procedury i funkcje w kodzie VBA
- potrafić wykonywać kod SQL z poziomu kodu VBA

Aby uczeń mógł otrzymać ocenę celującą powinien spełniać wymagania na ocenę bardzo dobrą oraz jedno z poniższych kryteriów:

- osiągnie sukces w konkursie związanym z tematyką zajęć
- wykona dodatkowy projekt, wykazując się dodatkowymi umiejętnościami, zgodny z nauczonym przedmiotem. Projekt ma być ustalony z nauczycielem i realizowany systematycznie, a postępy będą prezentowane nauczycielowi w ciągu całego roku szkolnego lub semestru
- osiągnie średnią ocen zgodną z WSO, klasyfikującą do wystawienia oceny celującej

PRZEDMIOT: PRACOWNIA STRON INTERNETOWYCH

program nr 351406 kwalifikacja inf03

Kryteria oceniania według skali ocen.

Na ocenę dopuszczającą uczeń powinien umieć:

- Uczeń stosuje podstawowe elementy języków skryptowych
- Potrafi zastosować poszczególne znaczniki
- Potrafi tworzyć proste strony oparte na znacznikach sekcji
- Tworzy nagłówki i paragrafy
- Stosuje elementy list, tabel, obrazów, odsyłaczy, formularzy z podstawowymi elementami
- Zapisuje zdarzenia i skrypty
- Tworzy style wewnętrzne i zewnętrzne
- Wykorzystuje kaskadowość stylów CSS
- Wykorzystuje różnego rodzaju selektory
- Stosuje reguły CSS (składnia stylu wewnętrznego zewnętrznego i inline)
- Wykorzystuje systemy zarządzania treścią
- Obsługuje panelu administracyjny CMS
- Konfiguruje system CMS
- Instaluje podstawowe rozszerzenia w systemie CMS
- Wykorzystuje gotowe szablony dla przerabianego CMS-a
- Dokonuje podstawowych operacje w edytorach Gimp i Inkscape.
- Zapisuje pliki graficzne w różnych formatach
- Wykorzystuje podstawowe modele barw w edytorze GIMP i INKSKAPE
- Obsługuje Corel Video Studio oraz SWISH Max
- Tworzy różne projekty graficzne strony internetowej
- Przypisuje barwy dla różnych elementów na stronie internetowej
- Tworzy fragmenty tekstu z różną wielkością czcionki na przykładowych stronach internetowych
- Testuje poprawność strony internetowej
- Waliduje wcześniej stworzone strony internetowe
- Pozycjonuje strony WWW

Na ocenę dostateczną uczeń powinien umieć:

- Uczeń stosuje przykłady języków skryptowych
- Potrafi zastosować poszczególne znaczniki
- Potrafi tworzyć proste strony oparte na znacznikach sekcji
- Tworzy nagłówki i paragrafy
- Stosuje elementy list, tabel, obrazów, odsyłaczy, formularzy
- Zapisuje zdarzenia i skrypty
- Tworzy proste formularze wraz z polami w standardzie HTML5
- Tworzy style wewnętrzne i zewnętrzne
- Wykorzystuje kaskadowość stylów CSS
- Wykorzystuje różnego rodzaju selektory
- Stosuje reguły CSS (składnia stylu wewnętrznego zewnętrznego i inline)
- Wykorzystuje systemy zarządzania treścią
- Obsługuje panelu administracyjny CMS
- Konfiguruje system CMS
- Instaluje podstawowe rozszerzenia w systemie CMS

- Wykorzystuje gotowe szablony dla przerabianego CMS-a
- Dokonuje podstawowych operacje w edytorach Gimp i Inkscape
- Zapisuje pliki graficzne w różnych formatach
- Wykorzystuje podstawowe modele barw w edytorze GIMP i INKSKAPE
- Obsługuje Corel Video Studio oraz SWISH Max
- Przetwarza dźwięk w Corel Video Studio
- Obrabia filmy w programie Corel Video Studio oraz tworzy animacje w SWISH Max
- Zapisuje filmy i animacje w różnych formatach
- Tworzy różne projekty graficzne strony internetowej
- Przypisuje barwy dla różnych elementów na stronie internetowej
- Tworzy fragmenty tekstu z różną wielkością czcionki na przykładowych stronach internetowych
- Testuje poprawność strony internetowej
- Waliduje wcześniej stworzone strony internetowe
- Pozycjonuje strony WWW

Na ocenę dobrą uczeń powinien umieć:

- Uczeń stosuje przykłady języków skryptowych
- Potrafi zastosować poszczególne znaczniki
- Potrafi tworzyć proste strony oparte na znacznikach sekcji
- Tworzy nagłówki i paragrafy
- Stosuje elementy list, tabel, obrazów, odsyłaczy, formularzy
- Zapisuje zdarzenia i skrypty
- Tworzy formularze wraz z polami w standardzie HTML5
- Tworzy style wewnętrzne i zewnętrzne
- Wykorzystuje kaskadowość stylów CSS
- Wykorzystuje różnego rodzaju selektory
- Stosuje reguły CSS (składnia stylu wewnętrznego zewnętrznego i inline)
- Wykorzystuje systemy zarządzania treścią
- Obsługuje panelu administracyjny CMS
- Konfiguruje system CMS
- Instaluje podstawowe rozszerzenia w systemie CMS
- Wykorzystuje gotowe szablony dla przerabianego CMS-a
- Dokonuje podstawowych operacje w edytorach Gimp i Inkscape.
- Zapisuje pliki graficzne w różnych formatach
- Wykorzystuje podstawowe modele barw w edytorze GIMP i INKSKAPE
- Obsługuje Corel Video Studio oraz SWISH Max
- Przetwarza dźwięk w Corel Video Studio
- Obrabia filmy w programie Corel Video Studio oraz tworzy animacje w SWISH Max
- Zapisuje filmy i animacje w różnych formatach
- Tworzy różne projekty graficzne strony internetowej
- Przypisuje barwy dla różnych elementów na stronie internetowej
- Tworzy fragmenty tekstu z różną wielkością czcionki na przykładowych stronach internetowych
- Potrafi dostosować strony do zasad i wytycznych dotyczących dostępności treści internetowych (WCAG 2.0)
- Testuje poprawność strony internetowej
- Waliduje wcześniej stworzone strony internetowe
- Pozycjonuje strony WWW

Na ocenę bardzo dobrą uczeń powinien umieć:

- Uczeń stosuje zaawansowane przykłady języków skryptowych
- Potrafi zastosować wszystkie znaczniki HTML-a
- Potrafi tworzyć zaawansowane strony oparte na znacznikach sekcji
- Tworzy nagłówki i paragrafy
- Definiuje elementy list, tabel, obrazów, odsyłaczy, formularzy, multimediiów
- Zapisuje zdarzenia i rozbudowane skrypty
- Tworzy formularze wraz z polami w standardzie HTML5
- Tworzy style wewnętrzne i zewnętrzne
- Wykorzystuje kaskadowość stylów CSS
- Wykorzystuje różnego rodzaju selektory
- Stosuje reguły CSS (składnia stylu wewnętrznego zewnętrznego i inline)
- Tworzy szablony stron wraz z stylem CSS
- Tworzy responsywny układ strony
- Obsługuje panel administracyjny CMS
- Konfiguruje system CMS
- Instaluje rozszerzenia w systemie CMS
- Wykorzystuje gotowe szablony dla przerabianego CMS-a
- Dokonuje zaawansowanych operacji w edytorach Gimp i Inkscape.
- Zapisuje pliki graficzne w różnych formatach
- Wykorzystuje różne modele barw w edytorze GIMP i INKSKAPE
- Obsługuje Corel Video Studio oraz SWISH Max
- Przetwarza dźwięk w Corel Video Studio
- Obrabia filmy w programie Corel Video Studio oraz tworzy animacje w SWISH Max
- Zapisuje filmy i animacje w różnych formatach
- Tworzy różne projekty graficzne strony internetowej
- Przypisuje barwy dla różnych elementów na stronie internetowej
- Tworzy fragmenty tekstu z różną wielkością czcionki na przykładowych stronach internetowych
- Potrafi dostosować strony do zasad i wytycznych dotyczących dostępności treści internetowych (WCAG 2.0)
- Tworzy strony dostosowane do osób niepełnosprawnych
- Testuje poprawność strony internetowej
- Waliduje wcześniej stworzone strony internetowe
- Pozycjonuje strony WWW
- Sprawdza czy strona spełnia wymagania w zakresie optymalizacji
- Wykorzystuje narzędzia do testowania szybkości wczytywania strony internetowej
- Dokonuje wyboru usług hostingu strony internetowej
- Wykonuje operacje na domenach internetowych
- Publikuje strony internetowe
- Wykorzystuje program FileZilla

WYMAGANIA WYKRACZAJĄCE

- Uczeń otrzymuje ocenę **celującą**, jeśli spełnia kryteria na ocenę bardzo dobrą oraz spełni przynajmniej jedno z kryteriów
- osiągnie sukces w konkursie związanym z tematyką zajęć
- wykona dodatkowy projekt, wykazując się dodatkowymi umiejętnościami, zgodny z nauczonym przedmiotem. Projekt ma być ustalony z nauczycielem i realizowany

systematycznie, a postępy będą prezentowane nauczycielowi w ciągu całego roku szkolnego lub semestru

- osiągnie średnią ocen zgodną z WSO, klasyfikującą do wystawienia oceny celującej

PROJEKTOWANIE I TWORZENIE STRON INTERNETOWYCH

program nr 351406 kwalifikacja inf03

Kryteria oceniania według skali ocen.

Na ocenę dopuszczającą uczeń powinien umieć:

- Uczeń zna podstawowe elementy języków skryptowych
- Zna poszczególne znaczniki
- Zna zasady tworzenia prostych stron opartych na znacznikach sekcji
- Zna metody tworzenia nagłówków i paragrafów
- Zna elementy listy, tabeli, obrazu, odsyłacza, formularza z podstawowymi elementami
- Posiada umiejętność tworzenia zdarzeń i skryptów
- Rozróżnia style wewnętrzne i zewnętrzne
- Zna pojęcie kaskadowości stylów CSS
- Potrafi wykorzystać różnego rodzaju selektory
- Zna reguły CSS (składnia stylu wewnętrznego zewnętrznego i inline)
- Zna systemy zarządzania treścią
- Posiada znajomość panelu administracyjnego CMS
- Zna zasady konfiguracji systemu CMS
- Zna sposoby instalacji podstawowych rozszerzeń w systemie CMS
- Wykorzystuje gotowe szablony dla przerabianego CMS-a
- Zna podstawowe operacje w edytorach Gimp i Inkscape.
- Potrafi zapisywać pliki graficzne w różnych formatach
- Zna podstawowe modele barw w edytorze GIMP i INKSKAPE
- Zna podstawy Corel Video Studio oraz SWISH Max
- Zna sposoby zapisu barw dla różnych elementów na stronie internetowej
- Zna sposoby testowania poprawności strony internetowej
- Zna sposoby walidacji wcześniej stworzonej strony internetowej
- Zna sposoby pozycjonowania strony WWW

Na ocenę dostateczną uczeń powinien umieć:

- Uczeń zna elementy języków skryptowych
- Zna poszczególne znaczniki
- Zna zasady tworzenia prostych stron opartych na znacznikach sekcji
- Zna metody tworzenia nagłówków i paragrafów
- Zna elementy listy, tabeli, obrazu, odsyłacza, formularza
- Posiada umiejętność tworzenia zdarzeń i skryptów
- Rozróżnia style wewnętrzne i zewnętrzne
- Zna zasady tworzenie prostych formularzy wraz z polami w standardzie HTML5
- Rozróżnia style wewnętrzne i zewnętrzne
- Zna pojęcie kaskadowości stylów CSS
- Potrafi wykorzystać różnego rodzaju selektory
- Zna reguły CSS (składnia stylu wewnętrznego zewnętrznego i inline)
- Zna systemy zarządzania treścią
- Posiada znajomość panelu administracyjnego CMS
- Zna zasady konfiguracji systemu CMS
- Zna sposoby instalacji rozszerzeń w systemie CMS
- Wykorzystuje gotowe szablony dla przerabianego CMS-a
- Zna podstawowe operacje w edytorach Gimp i Inkscape

- Potrafi zapisywać pliki graficzne w różnych formatach
- Zna modele barw w edytorze GIMP i INKSCAPE
- Zna podstawy Corel Video Studio oraz SWISH Max
- Wie jak przetwarzać dźwięk w Corel Video Studio
- Wie jak obrabiać filmy w programie Corel Video Studio oraz tworzyć animacje w SWISH Max
- Zna formy zapisu filmów i animacji w różnych formatach
- Wie jak tworzyć różne projekty graficzne stron internetowych
- Wie jak przypisać barwy dla różnych elementów na stronie internetowej
- Wie jak różnicować wielkość czcionki na przykładowych stronach internetowych
- Zna zasady testowania poprawności strony internetowej
- Zna zasady walidacji wcześniej stworzonej strony internetowej
- Zna zasady pozycjonowania strony WWW

Na ocenę dobrą uczeń powinien umieć:

- Uczeń zna elementy języków skryptowych
- Zna poszczególne znaczniki
- Zna metody tworzenia prostych stron opartych na znacznikach sekcji
- Zna zasady tworzenia nagłówków i paragrafów
- Zna elementy listy, tabeli, obrazu, odsyłacza, formularza
- Zna metody zapisu zdarzenia i skrypty
- Posiada umiejętność tworzenia formularzy wraz z polami w standardzie HTML5
- Wie jak tworzyć style wewnętrzne i zewnętrzne
- Zna pojęcie kaskadowości stylów CSS
- Wie jak wykorzystać różnego rodzaju selektory
- Zna reguły CSS (składnia stylu wewnętrznego zewnętrznego i inline)
- Wie jak wykorzystać systemy zarządzania treścią
- Zna metody obsługi panelu administracyjny CMS
- Wie jak konfigurować system CMS
- Posiada umiejętność instalowania rozszerzeń w systemie CMS
- Potrafi wykorzystywać gotowe szablony dla przerabianego CMS-a
- Potrafi dokonywać podstawowych operacji w edytorach Gimp i Inkscape
- Potrafi zapisywać pliki graficzne w różnych formatach
- Potrafi wykorzystać podstawowe modele barw w edytorze GIMP i INKSCAPE
- Obsługuje Corel Video Studio oraz SWISH Max
- Przetwarza dźwięk w Corel Video Studio
- Wie jak obrabiać filmy w programie Corel Video Studio oraz tworzyć animacje w SWISH Max
- Potrafi zapisywać filmy i animacje w różnych formatach
- Wie jak tworzyć różne projekty graficzne stron internetowych
- Wie jak przypisać barwy dla różnych elementów na stronie internetowej
- Wie jak różnicować wielkość czcionki na przykładowych stronach internetowych
- Wie jak dostosować strony do zasad i wytycznych dotyczących dostępności treści internetowych (WCAG 2.0)
- Zna metody sprawdzania poprawności strony internetowej
- Zna metody walidacji wcześniej stworzonej strony internetowej
- Zna metody pozycjonowania strony WWW

Na ocenę bardzo dobrą uczeń powinien umieć:

- Uczeń zna zaawansowane przykłady języków skryptowych
- Wie jak zastosować wszystkie znaczniki HTML-a

- Wie jak tworzyć zaawansowane strony oparte na znacznikach sekcji
- Wie jak tworzyć nagłówki i paragrafy
- Wie jak definiować elementy list, tabel, obrazów, odsyłaczy, formularzy, multimediów
- Zna zasady zapisu zdarzenia i rozbudowanego skryptu
- Wie jak tworzyć formularze wraz z polami w standardzie HTML5
- Wie jak tworzyć style wewnętrzne i zewnętrzne
- Zna zasady wykorzystania kaskadowości stylów CSS
- Wie jak wykorzystać różnego rodzaju selektory
- Zna reguły CSS (składnia stylu wewnętrznego zewnętrznego i inline)
- Wie jak tworzyć szablony stron wraz z stylem CSS
- Wie jak tworzyć responsywny układ strony
- Zna metody obsługi panelu administracyjnego CMS
- Zna zasady konfiguracji systemu CMS
- Zna zasady instalacji rozszerzeń w systemie CMS
- Wie jak zastosować gotowe szablony dla przerabianego CMS-a
- Zna zaawansowane operacje w edytorach Gimp i Inkscape
- Zapisuje pliki graficzne w różnych formatach
- Zna różne modele barw w edytorze GIMP i INKSKAPE
- Zna obsługę Corel Video Studio oraz SWISH Max
- Wie jak przetwarzać dźwięk w Corel Video Studio
- Zna zasady obróbki filmów w programie Corel Video Studio oraz tworzenia animacji w SWISH Max
- Zna metody zapisu filmów i animacji w różnych formatach
- Zna zasady tworzenia różnych projektów graficznych stron internetowych
- Wie jak przypisać barwy dla różnych elementów na stronie internetowej
- Zna metody różnicowania wielkości czcionki dla przykładowych stronach internetowych
- Wie jak dostosować strony do zasad i wytycznych dotyczących dostępności treści internetowych (WCAG 2.0)
- Wie jak tworzyć strony dostosowane do osób niepełnosprawnych
- Zna metody testowania poprawności strony internetowej
- Wie jak dokonać walidacji wcześniej stworzonej strony internetowej
- Zna metody pozycjonowania strony WWW
- Zna wymagania w zakresie optymalizacji strony
- Zna narzędzia do testowania szybkości wczytywania strony internetowej
- Potrafi wybrać usługi hostingu strony internetowej
- Wie jak wykonać operacje na domenach internetowych
- Zna zasady publikacji strony internetowe
- Zna obsługę programu FileZilla

WYMAGANIA WYKRACZAJĄCE

- Uczeń otrzymuje ocenę **celującą**, jeśli spełnia kryteria na ocenę bardzo dobrą oraz spełni przynajmniej jedno z kryteriów
- osiągnie sukces w konkursie związanym z tematyką zajęć
- wykona dodatkowy projekt, wykazując się dodatkowymi umiejętnościami, zgodny z nauczonym przedmiotem. Projekt ma być ustalony z nauczycielem i realizowany systematycznie, a postępy będą prezentowane nauczycielowi w ciągu całego roku szkolnego lub semestru
- osiągnie średnią ocen zgodną z WSO, klasyfikującą do wystawienia oceny celującej

PRZEDMIOT: PODSTAWY INFORMATYKI I BHP

Kryteria oceniania według skali ocen.

WYMAGANIA KONIECZNE

Uczeń otrzymuje ocenę **dopuszczającą**, jeśli potrafi:

- Wymienia obowiązujące przepisy prawa związane z bezpieczeństwem i higieną pracy, ochroną przeciwpożarową, ochroną środowiska i ergonomią
- Wymienia wewnątrzzakładowe przepisy prawa związane z bezpieczeństwem i higieną pracy, ochroną przeciwpożarową, ochroną środowiska i ergonomią
- Opisuje wymagania dotyczące ergonomii pracy w branży informatycznej
- Wymienia zadania i uprawnienia w zakresie ochrony pracy
- Wymienia zadania i uprawnienia w zakresie ochrony środowiska naturalnego;
- Wymienia i wskazuje obowiązki pracodawcy w zakresie bezpieczeństwa i higieny pracy
- Charakteryzuje funkcje odzieży ochronnej
- Opisuje podstawowe symptomy wskazujące na stany nagłego zagrożenia zdrowia i życia
- Zabezpiecza siebie, osobę poszkodowaną i miejsce wypadku
- Układa poszkodowanego w pozycji bezpiecznej
- Powiadamia odpowiednie służby
- Udziela pierwszej pomocy w urazowych stanach zagrożenia życia i zdrowia, takich jak krwotoki, zmiążdżenia, amputacje, złamania, oparzenia
- Udziela pierwszej pomocy w nieurazowych stanach zagrożenia życia i zdrowia, takich jak omdlenie, zawał, udar
- Wykonuje na fantomie resuscytację oddechowo-kръżeniową zgodnie z wytycznymi Polskiej i Europejskiej Rady Resuscytacji

Aby uczeń mógł otrzymać **ocenę dostateczną** powinien spełniać wymagania na ocenę dopuszczającą oraz:

- Stosuje wewnątrzzakładowe przepisy prawa związane z bezpieczeństwem i higieną pracy, ochroną przeciwpożarową, ochroną środowiska i ergonomią
- Wskazuje prawa i obowiązki pracownika, który uległ wypadkowi przy pracy, wynikające z przepisów prawa
- Ocenia wpływ czynników szkodliwych na zdrowie i bezpieczeństwo pracowników
- Dobiera środki ochrony indywidualnej w zależności od czynników występujących na stanowisku pracy
- Określa metody eliminacji niebezpiecznych źródeł i szkodliwych czynników występujących w procesach pracy
- Ocenia sytuację poszkodowanego na podstawie analizy obserwowanych u niego objawów

Aby uczeń mógł otrzymać ocenę dobrą powinien spełniać wymagania na ocenę dostateczną oraz:

- Stosuje klasyfikację podziału czynników występujących w środowisku pracy
- Określa skutki występowania czynników w środowisku pracy
- Określa zasady doboru środków ochrony indywidualnej i zbiorowej podczas wykonywania zadań zawodowych
- Określa ergonomiczne zasady organizacji pracy i stanowisk pracy

Aby uczeń mógł otrzymać ocenę bardzo dobrą powinien spełniać wymagania na ocenę dobrą oraz:

- Wymienia zadania i uprawnienia w zakresie ochrony środowiska pracy
- Wymienia zadania i uprawnienia w zakresie stosowania maszyn i urządzeń
- Ocenia prawidłowość doboru środków ochrony indywidualnej i zbiorowej do wykonywanych zadań zawodowych

- Przewiduje wpływ zmian wprowadzanych na poszczególnych etapach procesu produkcyjnego na poziom bezpieczeństwa i higieny pracy
- Wskazuje rodzaje świadczeń z tytułu wypadku przy pracy
- Wskazuje rodzaje świadczeń z tytułu choroby zawodowej
- Organizuje działania prewencyjne zapobiegające powstawaniu pożaru lub innego zagrożenia w przedsiębiorstwie
- Organizuje wybrane stanowisko pracy zgodnie z wymogami ergonomii, przepisami bezpieczeństwa i higieny pracy, ochrony przeciwpożarowej i ochrony środowiska

Aby uczeń mógł otrzymać ocenę celującą powinien spełniać wymagania na ocenę bardzo dobrą oraz jedno z poniższych kryteriów:

- osiągnie sukces w konkursie związanym z tematyką zajęć
- wykona dodatkowy projekt, wykazując się dodatkowymi umiejętnościami, zgodny z nauczonym przedmiotem. Projekt ma być ustalony z nauczycielem i realizowany systematycznie, a postępy będą prezentowane nauczycielowi w ciągu całego roku szkolnego lub semestru
- osiągnie średnią ocen zgodną z WSO, klasyfikującą do wystawienia oceny celującej

Podstawy informatyki

WYMAGANIA KONIECZNE

Uczeń otrzymuje ocenę dopuszczającą, jeśli potrafi:

- Identyfikuje parametry urządzeń techniki komputerowej
- Porównuje parametry tego samego typu urządzeń techniki komputerowej (np. dwóch kart graficznych, dwóch dysków twardych),
- Dobiera urządzenia techniki komputerowej zgodnie z wymaganiami technicznymi dla danego stanowiska (np. grafika komputerowego)
- Identyfikuje systemy informatyczne, podaje przykłady systemów informacji przetwarzanych elektronicznie, w tym: system PESEL, nabór elektroniczny do szkół, e-dziennik, system bankowości elektronicznej, profil zaufany
- Opisuje miejsca przechowywania informacji: serwer lokalny, chmurę, nośniki danych
- Opisuje działanie portali społecznościowych
- Określa zasady bezpiecznego korzystania z portali społecznościowych
- Wymienia dostępne udogodnienia dla osób z niepełno sprawnościami
- Wymienia topologie sieci
- Stosuje programy monitorujące łącze internetowe
- Wymienia i stosuje zasady bezpieczeństwa przy korzystaniu z sieci
- Stosuje zasadę etykiety
- Wykorzystuje dostępne narzędzia informatyczne do wykonywania działań na liczbach zapisanych w różnych pozycyjnych systemach liczbowych (np. kalkulatory HEX, DEC, BIN)
- Rozróżnia rodzaje szkodliwego oprogramowania
- Wymienia środki zabezpieczeń przed złośliwym oprogramowaniem oraz atakami
- Przestrzega zasad bezpiecznego przechowywania danych
- Przestrzega zasad bezpieczeństwa swojego cyfrowego wizerunku i tożsamości
- Korzysta ze źródeł informacji dotyczących norm i procedur oceny zgodności

Aby uczeń mógł otrzymać ocenę dostateczną powinien spełniać wymagania na ocenę dopuszczającą oraz:

- Identyfikuje cechy modelu TCP/IP i protokołów komunikacji sieciowej
- Opisuje sieć bezprzewodową oraz sieć przewodową
- Definiuje pojęcia: pobieranie i wysyłanie danych

- Używa komunikatorów tekstowych, audio/video oraz tablic interaktywnych
- Przekształca liczby zapisane w różnych pozycyjnych systemach liczbowych: dwójkowym, ósemkowym, szesnastkowym, dziesiętnym
- Rozróżnia rodzaje ataków hakerskich
- Przestrzega zasad prywatności w cyfrowym świecie
- Stosuje zasady dokonywania bezpiecznych transakcji w Internecie
- Wymienia cele normalizacji krajowej

Aby uczeń mógł otrzymać ocenę dobrą powinien spełniać wymagania na ocenę dostateczną oraz:

- Przelicza jednostki pojemności pamięci masowych
- Opisuje zasadę działania procesora i innych podzespołów
- Wymienia zależności między pamięcią operacyjną, procesorem i pozostałymi elementami systemu komputerowego
- Dobiera systemy informatyczne pod względem ich funkcjonalności
- Podaje przykłady zastosowań systemów informatycznych w działalności biznesowej, w tym: e-commerce, e-sklep, e-faktura, systemy rezerwacyjne
- Opisuje zasady działania sieci synchronicznej i asynchronicznej
- Wykonuje podstawowe działania logiczne i arytmetyczne na liczbach binarnych
- Wymienia zagrożenia dla sfery psychicznej (emocjonalnej), fizycznej, społecznej, poznawczej wynikające z przebywania w cyberprzestrzeni
- Wymienia i omawia podstawowe pojęcia związane z ochroną danych osobowych, informacji, prawami autorskimi i własnością intelektualną oraz wyjaśnia potrzebę ich ochrony

Aby uczeń mógł otrzymać ocenę bardzo dobrą powinien spełniać wymagania na ocenę dobrą oraz:

- Wymienia wymagania dotyczące poziomu dostępności według wytycznych WCAG 2.0
- Wskazuje różnice w działaniu sieci synchronicznej i asynchronicznej
- Zapisuje liczby w kodzie uzupełnieniowym do dwóch
- Opisuje zagrożenia dla sfery psychicznej (emocjonalnej), fizycznej, społecznej, poznawczej wynikające z przebywania w cyberprzestrzeni i sposoby przeciwdziałania tym zagrożeniom
- Wyjaśnia, czym jest norma i wymienia cechy normy
- Rozróżnia oznaczenie normy międzynarodowej, europejskiej i krajowej

Aby uczeń mógł otrzymać ocenę celującą powinien spełniać wymagania na ocenę bardzo dobrą oraz jedno z poniższych kryteriów:

- osiągnie sukces w konkursie związanym z tematyką zajęć
- wykona dodatkowy projekt, wykazując się dodatkowymi umiejętnościami, zgodny z nauczonym przedmiotem. Projekt ma być ustalony z nauczycielem i realizowany systematycznie, a postępy będą prezentowane nauczycielowi w ciągu całego roku szkolnego lub semestru
- osiągnie średnią ocen zgodną z WSO, klasyfikującą do wystawienia oceny celującej

PRZEDMIOT: PROGRAMOWANIE APLIKACJI INTERNETOWYCH

program nr 351406 kwalifikacja inf03

Kryteria oceniania według skali ocen.

Na ocenę dopuszczającą uczeń powinien umieć:

- charakteryzuje podstawowe pojęcia: aplikacja internetowa, skrypt, aplet
- opisuje reprezentacje algorytmów
- opisuje narzędzia programistyczne typu edytor, debugger, IDE
- opisuje strukturę aplikacji internetowej
- charakteryzuje skryptowe języki programowania
- charakteryzuje typy proste i złożone stosowane przy tworzeniu aplikacji
- charakteryzuje zmienne stosowane przy tworzeniu aplikacji
- charakteryzuje operatory stosowane przy tworzeniu aplikacji
- charakteryzuje instrukcje sterujące języków skryptowych
- charakteryzuje funkcje oraz wybrane biblioteki języków skryptowych
- opisuje funkcje języka JavaScript
- opisuje sposoby obsługi formularzy i kontrolek HTML
- potrafi scharakteryzować bibliotekę jQuery
- opisuje składnię języka PHP
- opisuje funkcje języka PHP
- opisuje funkcje obsługi plików
- opisuje sposoby przekazywania danych z formularza
- opisuje zastosowanie plików cookies
- opisuje zastosowanie sesji
- opisuje sposób współpracy aplikacji z bazą danych
- opisuje sposób definiowania zapytań do bazy danych
- charakteryzuje środowiska programistyczne aplikacji internetowych
- opisuje funkcje środowiska programistycznego
- analizuje błędy w kodzie źródłowym programu
- opisuje sposoby umieszczania komentarzy w kodzie źródłowym programu
- opisuje sposoby dokumentowania programu

Na ocenę dostateczną uczeń powinien umieć:

- charakteryzuje podstawowe pojęcia: aplikacja internetowa, skrypt, aplet
- opisuje reprezentacje algorytmów
- opisuje narzędzia programistyczne typu edytor, debugger, IDE
- opisuje strukturę aplikacji internetowej
- charakteryzuje skryptowe języki programowania
- charakteryzuje typy proste i złożone stosowane przy tworzeniu aplikacji
- charakteryzuje zmienne stosowane przy tworzeniu aplikacji
- charakteryzuje operatory stosowane przy tworzeniu aplikacji
- charakteryzuje instrukcje sterujące języków skryptowych
- charakteryzuje funkcje oraz wybrane biblioteki języków skryptowych
- opisuje funkcje języka JavaScript
- charakteryzuje obiektowość języka JavaScript
- opisuje obiekty przeglądarki
- opisuje obiekty języka JavaScript
- opisuje sposoby obsługi formularzy i kontrolek HTML

- charakteryzuje działanie wyrażeń regularnych
- potrafi scharakteryzować bibliotekę jQuery
- opisuje składnię języka PHP
- opisuje funkcje języka PHP
- opisuje funkcje obsługi plików
- opisuje sposoby przekazywania danych z formularza
- opisuje zastosowanie plików cookies
- opisuje zastosowanie sesji
- opisuje sposób współpracy aplikacji z bazą danych
- opisuje sposób definiowania zapytań do bazy danych
- charakteryzuje środowiska programistyczne aplikacji internetowych
- opisuje funkcje środowiska programistycznego
- analizuje błędy w kodzie źródłowym programu
- opisuje sposoby testowania tworzonych programów
- opisuje sposoby umieszczania komentarzy w kodzie źródłowym programu
- opisuje sposoby dokumentowania programu

Na ocenę dobrą uczeń powinien umieć:

- charakteryzuje podstawowe pojęcia: aplikacja internetowa, skrypt, aplet
- opisuje reprezentacje algorytmów
- opisuje narzędzia programistyczne typu edytor, debugger, IDE
- opisuje strukturę aplikacji internetowej
- charakteryzuje skryptowe języki programowania
- charakteryzuje typy proste i złożone stosowane przy tworzeniu aplikacji
- charakteryzuje zmienne stosowane przy tworzeniu aplikacji
- charakteryzuje operatory stosowane przy tworzeniu aplikacji
- charakteryzuje instrukcje sterujące języków skryptowych
- charakteryzuje funkcje oraz wybrane biblioteki języków skryptowych
- opisuje funkcje języka JavaScript
- charakteryzuje obiektowość języka JavaScript
- opisuje obiekty przeglądarki
- opisuje obiekty języka JavaScript
- opisuje obiekty dokumentu DOM
- charakteryzuje obsługę zdarzeń myszy i klawiatury
- opisuje sposoby obsługi formularzy i kontrolek HTML
- charakteryzuje działanie wyrażeń regularnych
- potrafi scharakteryzować bibliotekę jQuery
- opisuje składnię języka PHP
- opisuje funkcje języka PHP
- opisuje funkcje obsługi plików
- opisuje sposoby przekazywania danych z formularza
- opisuje zastosowanie plików cookies
- opisuje zastosowanie sesji
- opisuje sposób współpracy aplikacji z bazą danych
- opisuje sposób definiowania zapytań do bazy danych
- charakteryzuje środowiska programistyczne aplikacji internetowych
- opisuje funkcje środowiska programistycznego
- analizuje błędy w kodzie źródłowym programu
- opisuje sposoby testowania tworzonych programów
- opisuje sposoby umieszczania komentarzy w kodzie źródłowym programu

- opisuje sposoby dokumentowania programu
- opisuje elementy instrukcji użytkownika

Na ocenę bardzo dobrą uczeń powinien umieć:

- Uczeń zna zaawansowane przykłady języków skryptowych
- Wie jak zastosować wszystkie znaczniki HTML-a
- Wie jak tworzyć zaawansowane strony oparte na znacznikach sekcji
- Wie jak tworzyć nagłówki i paragrafy
- Wie jak definiować elementy list, tabel, obrazów, odsyłaczy, formularzy, multimediiów
- Zna zasady zapisu zdarzenia i rozbudowanego skryptu
- Wie jak tworzyć formularze wraz z polami w standardzie HTML5
- Wie jak tworzyć style wewnętrzne i zewnętrzne
- Zna zasady wykorzystania kaskadowości stylów CSS
- Wie jak wykorzystać różnego rodzaju selektory
- Zna reguły CSS (składnia stylu wewnętrznego zewnętrznego i inline)
- Wie jak tworzyć szablony stron wraz z stylem CSS
- Wie jak tworzyć responsywny układ strony
- Zna metody obsługi panelu administracyjnego CMS
- Zna zasady konfiguracji systemu CMS
- Zna zasady instalacji rozszerzeń w systemie CMS
- Wie jak zastosować gotowe szablony dla przerabianego CMS-a
- Zna zaawansowane operacje w edytorach Gimp i Inkscape
- Zapisuje pliki graficzne w różnych formatach
- Zna różne modele barw w edytorze GIMP i INKSKAPE
- Zna obsługę Corel Video Studio oraz SWISH Max
- Wie jak przetwarzać dźwięk w Corel Video Studio
- Zna zasady obróbki filmów w programie Corel Video Studio oraz tworzenia animacji w SWISH Max
- Zna metody zapisu filmów i animacji w różnych formatach
- Zna zasady tworzenia różnych projektów graficznych stron internetowych
- Wie jak przypisać barwy dla różnych elementów na stronie internetowej
- Zna metody różnicowania wielkości czcionki dla przykładowych stronach internetowych
- Wie jak dostosować strony do zasad i wytycznych dotyczących dostępności treści internetowych (WCAG 2.0)
- Wie jak tworzyć strony dostosowane do osób niepełnosprawnych
- Zna metody testowania poprawności strony internetowej
- Wie jak dokonać walidacji wcześniej stworzonej strony internetowej
- Zna metody pozycjonowania strony WWW
- Zna wymagania w zakresie optymalizacji strony
- Zna narzędzia do testowania szybkości wczytywania strony internetowej
- Potrafi wybrać usługi hostingu strony internetowej
- Wie jak wykonać operacje na domenach internetowych
- Zna zasady publikacji strony internetowej
- Zna obsługę programu FileZilla

WYMAGANIA WYKRACZAJĄCE

- Uczeń otrzymuje ocenę **celującą**, jeśli spełnia kryteria na ocenę bardzo dobrą oraz spełni przynajmniej jedno z kryteriów
- osiągnie sukces w konkursie związanym z tematyką zajęć

Wymagania edukacyjne dla zawodu technik programista 351406

- wykona dodatkowy projekt, wykazując się dodatkowymi umiejętnościami, zgodny z nauczaniem przedmiotem. Projekt ma być ustalony z nauczycielem i realizowany systematycznie, a postępy będą prezentowane nauczycielowi w ciągu całego roku szkolnego lub semestru
- osiągnie średnią ocen zgodną z WSO, klasyfikującą do wystawienia oceny celującej

PRACOWNIA APLIKACJI INTERNETOWYCH

program nr 351406 kwalifikacja inf03

Kryteria oceniania według skali ocen.

Na ocenę dopuszczającą uczeń powinien umieć:

- charakteryzuje podstawowe pojęcia: aplikacja internetowa, skrypt, aplet
- potrafi zastosować algorytm do opisu problemu programistycznego
- stosuje narzędzia programistyczne
- potrafi zastosować język JavaScript do tworzenia aplikacji internetowych
- stosuje język PHP do tworzenia aplikacji internetowych
- posługuje się typami prostymi i złożonymi oraz zmiennymi przy tworzeniu aplikacji w językach JavaScript, PHP oraz C#
- posługuje się operatorami przy tworzeniu aplikacji w językach JavaScript, PHP oraz C#
- potrafi zastosować instrukcje sterujące języków skryptowych
- potrafi zastosować funkcje oraz wybrane biblioteki języków skryptowych
- tworzy strony internetowe wykorzystujące skryptowe języki programowania
- stosuje funkcje języka JavaScript
- stosuje skrypty obsługujące formularze i kontrolki HTML
- stosuje skrypt PHP w kodzie HTML
- potrafi zastosować funkcje języka PHP
- tworzy skrypty przekazujące dane z formularza
- definiuje połączenie z bazą danych
- definiuje zapytania do bazy danych
- tworzy programy w wybranym środowisku programistycznym (XAMPP, Visual Studio)
- instaluje i konfiguruje serwer WWW (Apache, ASP.NET)
- instaluje i konfiguruje serwer MS SQL
- instaluje i konfiguruje serwer MySQL
- korzysta z pakietu phpMyAdmin
- analizuje błędy w kodzie źródłowym programu
- poprawia błędy w tworzonych programach
- stosuje debugger w przeglądarce Internetowej
- potrafi zastosować komentarze w kodzie źródłowym programu

Na ocenę dostateczną uczeń powinien umieć:

- charakteryzuje podstawowe pojęcia: aplikacja internetowa, skrypt, aplet
- potrafi zastosować algorytm do opisu problemu programistycznego
- stosuje narzędzia programistyczne
- potrafi zastosować język JavaScript do tworzenia aplikacji internetowych
- stosuje język PHP do tworzenia aplikacji internetowych
- posługuje się typami prostymi i złożonymi oraz zmiennymi przy tworzeniu aplikacji w językach JavaScript, PHP oraz C#
- posługuje się operatorami przy tworzeniu aplikacji w językach JavaScript, PHP oraz C#
- potrafi zastosować instrukcje sterujące języków skryptowych
- potrafi zastosować funkcje oraz wybrane biblioteki języków skryptowych
- tworzy strony internetowe wykorzystujące skryptowe języki programowania
- stosuje funkcje języka JavaScript
- potrafi zastosować obsługę zdarzeń myszy i klawiatury
- stosuje skrypty obsługujące formularze i kontrolki HTML

- stosuje walidację formularzy z wykorzystaniem biblioteki jQuery
- stosuje skrypt PHP w kodzie HTML
- potrafi zastosować funkcje języka PHP
- tworzy skrypty przekazujące dane z formularza
- tworzy skrypty wykorzystujące mechanizm sesji
- definiuje połączenie z bazą danych
- definiuje zapytania do bazy danych
- dobiera środowisko programistyczne do określonych zadań i języka programowania
- tworzy programy w wybranym środowisku programistycznym (XAMPP, Visual Studio)
- instaluje i konfiguruje serwer WWW (Apache, ASP.NET)
- instaluje i konfiguruje serwer MS SQL
- instaluje i konfiguruje serwer MySQL
- korzysta z pakietu phpMyAdmin
- korzysta z pakietu Management Studio
- analizuje błędy w kodzie źródłowym programu
- poprawia błędy w tworzonych programach
- stosuje debugger w przeglądarce Internetowej
- potrafi zastosować komentarze w kodzie źródłowym programu
- tworzy dokumentację programu

Na ocenę dobrą uczeń powinien umieć:

- charakteryzuje podstawowe pojęcia: aplikacja internetowa, skrypt, aplet
- potrafi zastosować algorytm do opisu problemu programistycznego
- stosuje narzędzia programistyczne
- potrafi zastosować język JavaScript do tworzenia aplikacji internetowych
- stosuje język PHP do tworzenia aplikacji internetowych
- posługuje się typami prostymi i złożonymi oraz zmiennymi przy tworzeniu aplikacji w językach JavaScript, PHP oraz C#
- posługuje się operatorami przy tworzeniu aplikacji w językach JavaScript, PHP oraz C#
- potrafi zastosować instrukcje sterujące języków skryptowych
- potrafi zastosować funkcje oraz wybrane biblioteki języków skryptowych
- tworzy strony internetowe wykorzystujące skryptowe języki programowania
- stosuje funkcje języka JavaScript
- stosuje obiekty przeglądarki
- potrafi zastosować obiekty dokumentu DOM
- potrafi zastosować obsługę zdarzeń myszy i klawiatury
- stosuje skrypty obsługujące formularze i kontrolki HTML
- stosuje obsługę zdarzeń myszy i klawiatury za pomocą biblioteki jQuery
- stosuje walidację formularzy z wykorzystaniem biblioteki jQuery
- stosuje skrypt PHP w kodzie HTML
- potrafi zastosować funkcje języka PHP
- stosuje funkcje obsługi plików
- tworzy skrypty przekazujące dane z formularza
- tworzy pliki cookies
- tworzy skrypty wykorzystujące mechanizm sesji
- definiuje połączenie z bazą danych
- definiuje zapytania do bazy danych
- dobiera środowisko programistyczne do określonych zadań i języka programowania
- tworzy programy w wybranym środowisku programistycznym (XAMPP, Visual Studio)
- instaluje i konfiguruje serwer WWW (Apache, ASP.NET)

- instaluje i konfiguruje serwer MS SQL
- instaluje i konfiguruje serwer MySQL
- korzysta z pakietu phpMyAdmin
- korzysta z pakietu Management Studio
- analizuje błędy w kodzie źródłowym programu
- wykonuje testy tworzonych programów
- poprawia błędy w tworzonych programach
- stosuje debugger w przeglądarce Internetowej
- potrafi zastosować komentarze w kodzie źródłowym programu
- tworzy dokumentację programu
- tworzy instrukcję użytkownika programu

Na ocenę bardzo dobrą uczeń powinien umieć:

- charakteryzuje podstawowe pojęcia: aplikacja internetowa, skrypt, aplet
- potrafi zastosować algorytm do opisu problemu programistycznego
- stosuje narzędzia programistyczne
- potrafi zastosować język JavaScript do tworzenia aplikacji internetowych
- stosuje język PHP do tworzenia aplikacji internetowych
- stosuje język C# do tworzenia aplikacji internetowych
- posługuje się typami prostymi i złożonymi oraz zmiennymi przy tworzeniu aplikacji w językach JavaScript, PHP oraz C#
- posługuje się operatorami przy tworzeniu aplikacji w językach JavaScript, PHP oraz C#
- potrafi zastosować instrukcje sterujące języków skryptowych
- potrafi zastosować funkcje oraz wybrane biblioteki języków skryptowych
- tworzy strony internetowe wykorzystujące skryptowe języki programowania
- stosuje funkcje języka JavaScript
- tworzyć obiekty w języku JavaScript
- stosuje obiekty przeglądarki
- stosuje obiekty języka JavaScript
- potrafi zastosować obiekty dokumentu DOM
- potrafi zastosować obsługę zdarzeń myszy i klawiatury
- stosuje skrypty obsługujące formularze i kontrolki HTML
- potrafi zastosować wyrażenia regularne do walidacji formularza
- stosuje obsługę zdarzeń myszy i klawiatury za pomocą biblioteki jQuery
- stosuje walidację formularzy z wykorzystaniem biblioteki jQuery
- tworzy skrypty z wykorzystaniem biblioteki AngularJS
- tworzyć skrypty z wykorzystaniem biblioteki React
- stosuje skrypt PHP w kodzie HTML
- potrafi zastosować funkcje języka PHP
- stosuje funkcje obsługi plików
- tworzy skrypty przekazujące dane z formularza
- tworzy pliki cookies
- tworzy skrypty wykorzystujące mechanizm sesji
- definiuje połączenie z bazą danych
- definiuje zapytania do bazy danych
- potrafi zastosować bibliotekę PDO do obsługi bazy Danych
- dobiera środowisko programistyczne do określonych zadań i języka programowania
- tworzy programy w wybranym środowisku programistycznym (XAMPP, Visual Studio)
- instaluje i konfiguruje serwer WWW (Apache, ASP.NET)
- instaluje i konfiguruje serwer MS SQL

- instaluje i konfiguruje serwer MySQL
- korzysta z pakietu phpMyAdmin
- korzysta z pakietu Management Studio
- analizuje błędy w kodzie źródłowym programu
- wykonuje testy tworzonych programów
- poprawia błędy w tworzonych programach
- stosuje debugger w przeglądarce Internetowej
- potrafi zastosować komentarze w kodzie źródłowym programu
- tworzy dokumentację programu
- tworzy instrukcję użytkownika programu

WYMAGANIA WYKRACZAJĄCE

- Uczeń otrzymuje ocenę **celującą**, jeśli spełnia kryteria na ocenę bardzo dobrą oraz spełni przynajmniej jedno z kryteriów
- osiągnięcie sukces w konkursie związanym z tematyką zajęć
- wykona dodatkowy projekt, wykazując się dodatkowymi umiejętnościami, zgodny z nauczonym przedmiotem. Projekt ma być ustalony z nauczycielem i realizowany systematycznie, a postępy będą prezentowane nauczycielowi w ciągu całego roku szkolnego lub semestru
- osiągnięcie średnią ocen zgodną z WSO, klasyfikującą do wystawienia oceny celującej

PROGRAMOWANIE APLIKACJI DESKTOPOWYCH

program nr 351406 kwalifikacja inf04

Uczeń potrafi:

1. scharakteryzować środowisko programistyczne (np. Visual Studio, CodeBlocks),
2. dobrać środowisko do programowania w wybranym języku (C++, C#, Python, Java, Visual Basic),
3. rozpoznać narzędzia wykorzystywane w procesie tworzenia aplikacji desktopowych
4. stosować typy proste: liczbowe stałe- i zmiennoprzecinkowe, logiczny, znakowy i łańcuchowy (string),
5. deklarować zmienne,
6. wczytać i wyświetlić wartość zmiennej,
7. konstruować wyrażenia arytmetyczne i logiczne z uwzględnieniem hierarchii operatorów,
8. wykorzystywać w programach tablice jedno- i dwuwymiarowe,
9. wykonać operacje na zmiennych typu prostego,
10. definiować i stosować typ rekordowy (struktura, unia),
11. zastosować typ plikowy do obsługi plików,
12. wykorzystywać w programach tablice dynamiczne i asocjacyjne,
13. zastosować typ wskaźnikowy i zmienne dynamicznie,
14. zastosować kolekcje (stosy, kolejki, listy, wektory),
15. dobrać typ zmiennej do reprezentowania wartości w programie,
16. wykonać w programie operacje przypisania, arytmetyczne, porównania, logiczne, bitowe, obsługi łańcuchów,
17. stosować rozgałęzienia w programie za pomocą instrukcji warunkowych if, [switch],
18. realizować powtórzenia w programie za pomocą pętli for, while, [do while],
19. programować wykorzystując wybrane biblioteki języka C++, C#, Python lub innego języka programowania: biblioteka standardowa, biblioteka z funkcjami matematycznymi, biblioteka z podstawowymi algorytmami
20. kompilować i uruchamiać programy,
21. analizować błędy w kodzie za pomocą debuggera,
22. tworzyć program z podziałem na funkcje (metody),
23. implementować algorytmy w programie,
24. definiować funkcje rekurencyjne,
25. zastosować rekurencję do realizacji powtórzeń,
26. stosować algorytm wyszukiwania dla różnych zestawów danych (tablic, list, kolejek, stosów),
27. objaśnić pojęcie frameworka,
28. zastosować framework Qt do tworzenia aplikacji desktopowych w języku C++ (framework WPF – do tworzenia aplikacji desktopowych w języku C#),
29. projektować interfejs użytkownika i wygląd aplikacji wykorzystując odpowiednie elementy (widżety, np. okna dialogowe, przyciski, paski narzędziowe),
30. projektować funkcjonalność aplikacji,
31. planować system zabezpieczeń aplikacji,
32. projektować struktury danych dla aplikacji,

33. odzwierciedlać rzeczywistość w formie zbioru obiektów,
34. definiować klasy i tworzyć obiekty,
35. definiować składowe klasy – pola i metody,
36. dobrać specyfikatory dostępu (public, private) dla składowych klasy,
37. definiować konstruktory, w tym kopiujące i destruktory,
38. odwoływać się poprzez obiekt do składowych klasy,
39. wykorzystywać w programach składowe statyczne klasy,
40. tworzyć funkcje zaprzyjaźnione z klasą,
41. tworzyć klasy zaprzyjaźnione,
42. tworzyć klasę pochodną do danej klasy,
43. projektować program z zastosowaniem hierarchii dziedziczenia klas,
44. określić przynależność składowych klas (pól i metod) do odpowiednich klas w hierarchii dziedziczenia,
45. definiować konstruktory klas pochodnych,
46. definiować klasy bazowe i pochodne,
47. wykorzystać mechanizm przeciążania (przeładowania) metod klasy,
48. definiować metody wirtualne,
49. tworzyć klasy abstrakcyjne,
50. zastosować w programie hermetyzację, dziedziczenie, polimorfizm,
51. wyjaśnić rolę szablonów klas w programowaniu obiektowym,
52. opisywać szablony klas dla prostych typów liczbowych,
53. stosować mechanizm obsługi wyjątków z instrukcjami try i catch,
54. zgłaszać wyjątki do obsłużenia (instrukcja throw),
55. sklasyfikować możliwe błędy wykonania aplikacji,
56. zrealizować obsługę błędów wykonania aplikacji,
57. stosować strategie organizowania okien w ramach pulpitu (SDI, MDI),
58. zastosować funkcje jednego z języków C++, C#, Java Python do tworzenia aplikacji desktopowych,
59. zainstalować oprogramowanie do tworzenia GUI, np. Qt Creator,
60. tworzyć okno aplikacji np. z wykorzystaniem Qt Quick,
61. zaprojektować interfejs użytkownika,
62. stworzyć projekt i strukturę aplikacji,
63. stworzyć okno aplikacji z ikoną programu, przyciskami maksymalizuj, minimalizuj,
64. wstawić podstawowe widżety (kontrolki), np.: pole edycji, suwak, przycisk do okna aplikacji,
65. rozmieszczyć widżety w oknie aplikacji,
66. przypisać widżetom zasady funkcjonowania,
67. zaprojektować obsługę zdarzeń myszy i klawiatury,
68. zaprogramować obsługę zdarzeń myszy i klawiatury,
69. wstawić do aplikacji kod źródłowy,
70. stosować predefiniowane okna dialogowe (np. wybór pliku, czcionki, koloru),
71. zaprojektować i dodać menu do aplikacji,
72. implementować opcje menu np. Plik, Edycja,
73. utworzyć interfejs użytkownika,
74. stworzyć klasyczne okno aplikacji desktopowej (pasek menu, pasek narzędzi, obszar dokumentu, pasek statusu)

75. stosować język do projektowania interfejsu użytkownika, np. QML, w tym np.: stosować podstawowe elementy wizualne i niewizualne, wykonać transformację obiektu (np. przesunięcie, obrót, skalowanie), pozycjonować elementy, animować właściwości elementów, np.: kolor, obrót, przejście,
76. zaprojektować własne okna dialogowe,
77. utworzyć okienkowe aplikacje desktopowe, np. przeglądarka zdjęć, notatnik,
78. korzystać z elementów multimedialnych w aplikacji,
79. przechowywać dane aplikacji,
80. dynamicznie ładować elementy,
81. dobrać narzędzia i środowisko do testowania aplikacji,
82. dokonać statycznej analizy kodu w celu znalezienia nieefektywnych konstrukcji oraz fragmentów kodu,
83. przeprowadzić testy programów,
84. usunąć błędy i niedoskonałości ze swoich kodów źródłowych,
85. optymalizować kod źródłowy,
86. umieścić komentarze w kodzie źródłowym programu,
87. stosować w kodzie źródłowym komentarze typu DocBlocks,
88. tworzyć dokumentację techniczną aplikacji,
89. tworzyć pliki pomocy,
90. tworzyć instrukcję użytkownika programu,
91. opracować dokumentację wdrożenia projektu,
92. opracować dokumentację z przeprowadzonych testów aplikacji,
93. dokonać podziału testów (np. względu na weryfikowane obiekty, ze względu na metodę weryfikacji, bazujące na wymaganiach),
94. zaplanować fazy testowania,
95. przeprowadzać testy w kolejnych fazach projektu informatycznego, przeprowadzać testy funkcjonalne,
96. stosować narzędzia do automatyzacji procesu testowania,
97. korzystać z systemów raportowania błędów, np. BugZilla, JIRA,
98. przeprowadzać testy interfejsu,
99. testować prototyp projektu interfejsu,
100. przeprowadzać testy нефункционалне: użyteczności, wydajnościowe, obciążeniowe, zgodności, bezpieczeństwa,
101. przygotować środowisko testowe,
102. tworzyć scenariusze testowania aplikacji,
103. aktywnie słuchać ,włączając się do dyskusji podczas szukania sposobu rozwiązania problemu,
104. stosować zasady kultury osobistej i ogólnie przyjęte normy zachowania w swoim środowisku;
105. zastosować właściwą technikę twórczego myślenia przy rozwiązaniu problemu;
106. wykazywać się kreatywnością w rozwiązywaniu problemów,
107. ponosić odpowiedzialność za podejmowane działania,
108. dokonać analizy i oceny podejmowanych działań,
109. doskonalić jakość wykonywanych działań;
110. doskonalić umiejętności zawodowe,

111. pracować w zespole.

Ocena celująca:

1. Uczeń spełnia wymagania na ocenę bardzo dobrą i samodzielnie poszerza swoją wiedzę z zakresu przedmiotów zawodowych.
2. Uczeń wykazuje zainteresowanie problematyką przedmiotową, a jego zaangażowanie potwierdzone jest aktywnością na zajęciach i poza nimi.
3. Uczeń wykonuje dodatkowe zadania. Pomysłowo i oryginalnie rozwiązuje nietypowe zadania;
4. Uczeń odnosi sukcesy w konkursach, olimpiadach, itp.

Ocena bardzo dobra:

1. Uczeń opanował pełny zakres wiedzy i umiejętności określonych programem nauczania .
2. Uczeń sprawnie posługuje się zdobytymi wiadomościami, potrafi ocenić sytuację, wyciągać prawidłowe wnioski.
3. Uczeń sprawnie wykonuje wszystkie zadania.

Ocena dobra:

1. Uczeń dobrze opanował wiedzę i umiejętności określone programem nauczania.
2. Uczeń dysponuje dobrą umiejętnością zastosowania zdobytych wiadomości.
3. Uczeń uzyskuje stałe dobre postępy podczas zajęć, bierze aktywny udział w lekcji.

Ocena dostateczna:

1. Uczeń opanował wiadomości i umiejętności umożliwiające zdobywanie dalszej wiedzy.
2. Uczeń dysponuje przeciętną wiedzą w zakresie materiału przewidzianego programem nauczania, w jego wiadomościach są luki.
3. Uczeń nie wykazuje większego zainteresowania przedmiotem.
4. Wykonywanie zadań przez ucznia budzi zastrzeżenia.

Ocena dopuszczająca:

1. Zdobyte wiadomości są niewystarczające na uzyskanie przez ucznia podstawowej wiedzy z przedmiotu.
2. Uczeń zadania o niewielkim stopniu trudności rozwiązuje przy pomocy nauczyciela.
3. Uczeń niechętnie bierze udział w lekcjach.
4. Uczeń ma lekceważący stosunek do przedmiotu

Ocena niedostateczna:

1. Uczeń nie opanował wiadomości i umiejętności, jakie były nabywane w danej klasie, braki w wiadomościach i umiejętnościach uniemożliwiają dalsze zdobywanie wiedzy;
2. Uczeń nie potrafi rozwiązywać zadań o niewielkim stopniu trudności.

PRACOWNIA PROGRAMOWANIA APLIKACJI DESKTOPOWYCH

program nr 351406 kwalifikacja inf04

Uczeń potrafi:

1. dobrać środowisko programistyczne do określonych zadań i języka programowania (np. Visual Studio) ,
2. opisać narzędzia wykorzystywane w procesie tworzenia aplikacji desktopowych,
3. objaśnić pojęcie frameworka,
4. opisać funkcje frameworka do tworzenia aplikacji desktopowych np. Qt w języku C++ (lub frameworka WPF – w języku C#)
5. opisać strategie organizowania okien w ramach pulpitu (SDI, MDI)
6. opisać rodzaje widżetów (kontrolki),
7. zaprojektować interfejs użytkownika,
8. stworzyć projekt i strukturę aplikacji,
9. zaprojektować okna aplikacji,
10. opisać zasady dodawania widżetów do okna aplikacji i sposoby ich rozmieszczania,
11. zaprojektować obsługę zdarzeń myszy i klawiatury,
12. zaprojektować menu aplikacji,
13. zaprojektować okna dialogowe aplikacji,
14. scharakteryzować język do projektowania interfejsu użytkownika, np. QML,
15. określić funkcje narzędzi do zarządzania projektem ,
16. dobrać narzędzia do zarządzania projektem,
17. zastosować narzędzia wizualizacji w zarządzaniu etapami projektu, zadaniami i czasem, np. diagram Gantta
18. korzystać z programów wspierających zarządzanie projektami, np. Jira, Trello,
19. korzystać z systemu kontroli wersji, np. Git,
20. analizować wymagania klienta i tworzyć aplikację zgodnie z nimi,
21. zidentyfikować elementy interfejsu użytkownika, np. formularze, paski narzędziowe, widżety,
22. zaprojektować interfejs użytkownika i wygląd aplikacji,
23. zaprojektować struktury danych dla aplikacji,
24. zaprojektować funkcjonalność aplikacji
25. utworzyć specyfikację techniczną dla zespołu programistów na podstawie wymagań klienta,
26. dostosować interfejs aplikacji do różnych platform,
27. zaprojektować aplikacje zgodnie z paradygmatami programowania: strukturalnym, obiektywnym,
28. zaprojektować aplikację opartą na architekturze klient-serwer
29. zaplanować system zabezpieczeń aplikacji,
30. określić cel projektu ,
31. określić fazy realizacji projektu,
32. scharakteryzować cykl życia projektu informatycznego i jego poszczególne etapy,
33. określić zasoby ludzkie, ramy czasowe i koszt wykonania projektu,
34. planować etapy tworzenia aplikacji,

35. zastosować metodologię zarządzania projektem: model kaskadowy (waterfall), model przyrostowy, model prototypowy, metodyki zwinne (Agile oraz przynajmniej jedną z Scrum, Lean, Kanban) ,
36. dobrać optymalną metodologię zarządzania projektem,
37. organizować prace projektowe (wyznaczyć role, ustalić etapy prac),
38. stosować harmonogram czynności w celu efektywnego osiągnięcia celów,
39. dobrać wzorzec projektowy do zadania programistycznego,
40. stosować wzorce projektowe w programowaniu obiektowym, np. Metoda szablonowa (Template method), Fasada (Facade), Kompozyt (Composite)
41. rozróżniać autorskie prawa osobiste i majątkowe,
42. określić czas trwania praw autorskich,
43. określić konsekwencje naruszenia prawa autorskiego,
44. rozróżniać typy licencji oprogramowania,
45. scharakteryzować elementy własności intelektualnej (dobra niematerialne, własności przemysłowe),
46. rozróżnić dedykowane języki do programowania aplikacji mobilnych tj, Objective-C, Swift, Java, C#,
47. scharakteryzować typy danych w wybranym języku programowania,
48. deklarować zmienne w wybranym języku programowania,
49. deklarować tablice jedno i wielowymiarowe w wybranym języku programowania,
50. deklarować struktury danych w wybranym języku programowania,
51. wykonywać operacje arytmetyczne, bitowe, logiczne, przypisania, porównania na zmiennych w wybranym języku programowania,
52. obsługiwać operacje wejścia/wyjścia w wybranym języku programowania,
53. przetwarzać ciągi znaków i formatować dane tekstowe w wybranym języku programowania,
54. dobrać operatory warunkowe w wybranym języku programowania,
55. podać implementację instrukcji jeżeli w wybranym języku programowania,
56. podać implementację instrukcji przełączającej w wybranym języku programowania,
57. podać implementację poszczególnych pętli w wybranym języku programowania,
58. podać implementację instrukcji sterujących programem w wybranym języku programowania,
59. deklarować klasy i metody w wybranym języku programowania,
60. dobrać typy argumentów w metodach w wybranym języku programowania,
61. przeciążać metody w wybranym języku programowania,
62. definiować własne konstruktory i destruktory w klasie w wybranym języku programowania,
63. uzasadnić potrzebę stosowania hermetyzacji w programowaniu obiektowym,
64. uzasadnić potrzebę zastosowania dziedziczenia w wybranym języku programowania,
65. implementować klasy potomne w wybranym języku programowania,
66. implementować obsługę plików w wybranym języku programowania,
67. obsługiwać wyjątki i błędy w wybranym języku programowania,
68. dobrać środowiska programistyczne aplikacji mobilnych,
69. scharakteryzować narzędzia programistycznego środowiska aplikacji mobilnych,
70. określić parametry konfiguracji środowiska programistycznego aplikacji mobilnych,
71. scharakteryzować interakcje pomiędzy użytkownikiem i aplikacją mobilną,
72. rozróżnić gesty użytkownika dostępne w wybranym systemie iOS lub Android,

73. zidentyfikować elementy UI takie jak: przyciski, nawigacja, okna dialogowe, listy, formularze, paski narzędziowe, grafika, animacje, dźwięk w aplikacji mobilnej w systemie iOS,
74. identyfikować elementy UI takie jak: przyciski, nawigacja, okna dialogowe, listy, formularze, paski narzędziowe, grafika, animacje, dźwięk w aplikacji w systemie Android,
75. zastosować składnię języka XAML,
76. rozróżniać poszczególne elementy interfejsu użytkownika w kodzie XAML,
77. dobrać atrybuty elementów interfejsu użytkownika w języku XAML,
78. programować interfejs użytkownika w języku XAML,
79. zidentyfikować komentarze jedno i wielowierszowe,
80. stosować zastosowanie komentarzy w kodzie programu,
81. pisać dokumentację kodu,
82. pisać dokumentację pomocy do programu,
83. pisać instrukcję pomocy dla użytkownika,
84. pisać dokumentację wdrożenia projektu programistycznego,
85. rozróżniać narzędzia umożliwiające testowanie kodu programu w poszczególnych środowiskach programistycznych,
86. rozróżniać systemy raportowania błędów,
87. identyfikować błędy w kodzie programu zgłaszane przez interpreter,
88. scharakteryzować metody testowania programu,
89. przygotować testy funkcjonalne i нефункционалне programu,
90. planować i organizować pracę zespołu w celu wykonania przydzielonych zadań,
91. dobrać osoby do wykonania przydzielonych zadań,
92. kierować wykonaniem przydzielonych zadań,
93. oceniać jakość wykonania przydzielonych zadań, aktywnie słuchać ,włączając się do dyskusji podczas szukania sposobu rozwiązania problemu,
94. stosować zasady kultury osobistej i ogólnie przyjęte normy zachowania w swoim środowisku;
95. zastosować właściwą technikę twórczego myślenia przy rozwiązaniu problemu;
96. wykazywać się kreatywnością w rozwiązywaniu problemów,
97. ponosić odpowiedzialność za podejmowane działania,
98. dokonać analizy i oceny podejmowanych działań,
99. doskonalić jakość wykonywanych działań;

Ocena celująca:

5. Uczeń spełnia wymagania na ocenę bardzo dobrą i samodzielnie poszerza swoją wiedzę z zakresu przedmiotów zawodowych.
6. Uczeń wykazuje zainteresowanie problematyką przedmiotową, a jego zaangażowanie potwierdzone jest aktywnością na zajęciach i poza nimi.
7. Uczeń wykonuje dodatkowe zadania. Pomysłowo i oryginalnie rozwiązuje nietypowe zadania;
8. Uczeń odnosi sukcesy w konkursach, olimpiadach, itp.

Ocena bardzo dobra:

4. Uczeń opanował pełny zakres wiedzy i umiejętności określonych programem nauczania .
5. Uczeń sprawnie posługuje się zdobytymi wiadomościami, potrafi ocenić sytuację, wyciągać prawidłowe wnioski.

6. Uczeń sprawnie wykonuje wszystkie zadania.

Ocena dobra:

4. Uczeń dobrze opanował wiedzę i umiejętności określone programem nauczania.
5. Uczeń dysponuje dobrą umiejętnością zastosowania zdobytych wiadomości.
6. Uczeń uzyskuje stałe dobre postępy podczas zajęć, bierze aktywny udział w lekcji.

Ocena dostateczna:

5. Uczeń opanował wiadomości i umiejętności umożliwiające zdobywanie dalszej wiedzy.
6. Uczeń dysponuje przeciętną wiedzą w zakresie materiału przewidzianego programem nauczania, w jego wiadomościach są luki.
7. Uczeń nie wykazuje większego zainteresowania przedmiotem.
8. Wykonywanie zadań przez ucznia budzi zastrzeżenia.

Ocena dopuszczająca:

5. Zdobyte wiadomości są niewystarczające na uzyskanie przez ucznia podstawowej wiedzy z przedmiotu.
6. Uczeń zadania o niewielkim stopniu trudności rozwiązuje przy pomocy nauczyciela.
7. Uczeń niechętnie bierze udział w lekcjach.
8. Uczeń ma lekceważący stosunek do przedmiotu

Ocena niedostateczna:

3. Uczeń nie opanował wiadomości i umiejętności, jakie były nabywane w danej klasie, braki w wiadomościach i umiejętnościach uniemożliwiają dalsze zdobywanie wiedzy;
4. Uczeń nie potrafi rozwiązywać zadań o niewielkim stopniu trudności.

PROJEKTOWANIE OPROGRAMOWANIA

program nr 351406 kwalifikacja inf04

Uczeń potrafi:

1. rozróżniać kompilatory i interpretery,
2. scharakteryzować zadania kompilatora, interpretera, debuggera,
3. scharakteryzować pojęcie biblioteki,
4. scharakteryzować etapy kompilacji i interpretacji kodu,
5. zaprojektować algorytmy za pomocą różnych metod: schematów blokowych, listy kroków, drzew decyzyjnych, pseudokodu,
6. scharakteryzować algorytmy iteracyjne,
7. scharakteryzować metody sortowania i ich złożoność obliczeniową
8. podać przykłady operacji na zmiennych: wejścia i wyjścia, arytmetyczne, logiczne,
9. określić składnię instrukcji warunkowej i wyboru,
10. określić składnię instrukcji pętli,
11. podać definicję funkcji i opisać jej prototyp,
12. zapisać algorytmy w języku programowania,
13. objaśnić pojęcie rekurencji,
14. zastosować paradygmaty programowania obiektowego,
15. wyjaśnić pojęcia klasa, obiekt, metoda, pole, konstruktor, destruktor, dziedziczenie, hermetyzacja, polimorfizm,
16. sklasyfikować możliwe błędy wykonania aplikacji,
17. określić obsługę błędów wykonania aplikacji,
18. scharakteryzować algorytmy tekstowe i szyfrowania, tablicowe,
19. scharakteryzować algorytmy rekurencyjne,
20. zapisać w języku programowania różne algorytmy sortowania, np. bąbelkowe, przez wstawianie, zachłanne, szybkie, metodą dziel i zwyciężaj
21. scharakteryzować algorytmy np. heurystyczne, problem komiwojażera,
22. określić złożoność obliczeniową algorytmów,
23. ocenić efektywność różnych algorytmów sortowania,
24. omówić algorytmy wyszukiwania dla różnych typów danych.

Ocena celująca:

9. Uczeń spełnia wymagania na ocenę bardzo dobrą i samodzielnie poszerza swoją wiedzę z zakresu przedmiotów zawodowych.
10. Uczeń wykazuje zainteresowanie problematyką przedmiotową, a jego zaangażowanie potwierdzone jest aktywnością na zajęciach i poza nimi.
11. Uczeń wykonuje dodatkowe zadania. Pomysłowo i oryginalnie rozwiązuje nietypowe zadania;
12. Uczeń odnosi sukcesy w konkursach, olimpiadach, itp.

Ocena bardzo dobra:

7. Uczeń opanował pełny zakres wiedzy i umiejętności określonych programem nauczania .

8. Uczeń sprawnie posługuje się zdobytymi wiadomościami, potrafi ocenić sytuację, wyciągać prawidłowe wnioski.
9. Uczeń sprawnie wykonuje wszystkie zadania.

Ocena dobra:

7. Uczeń dobrze opanował wiedzę i umiejętności określone programem nauczania.
8. Uczeń dysponuje dobrą umiejętnością zastosowania zdobytych wiadomości.
9. Uczeń uzyskuje stałe dobre postępy podczas zajęć, bierze aktywny udział w lekcji.

Ocena dostateczna:

9. Uczeń opanował wiadomości i umiejętności umożliwiające zdobywanie dalszej wiedzy.
10. Uczeń dysponuje przeciętną wiedzą w zakresie materiału przewidzianego programem nauczania, w jego wiadomościach są luki.
11. Uczeń nie wykazuje większego zainteresowania przedmiotem.
12. Wykonywanie zadań przez ucznia budzi zastrzeżenia.

Ocena dopuszczająca:

9. Zdobyte wiadomości są niewystarczające na uzyskanie przez ucznia podstawowej wiedzy z przedmiotu.
10. Uczeń zadania o niewielkim stopniu trudności rozwiązuje przy pomocy nauczyciela.
11. Uczeń niechętnie bierze udział w lekcjach.
12. Uczeń ma lekceważący stosunek do przedmiotu

Ocena niedostateczna:

5. Uczeń nie opanował wiadomości i umiejętności, jakie były nabywane w danej klasie, braki w wiadomościach i umiejętnościach uniemożliwiają dalsze zdobywanie wiedzy;
6. Uczeń nie potrafi rozwiązywać zadań o niewielkim stopniu trudności.

PRACOWNIA PROJEKTOWANIA OPROGRAMOWANIA

program nr 351406 kwalifikacja inf04

Uczeń potrafi:

1. rozróżniać kompilatory i interpretery,
2. scharakteryzować zadania kompilatora, interpretera, debuggera,
3. scharakteryzować pojęcie biblioteki,
4. scharakteryzować etapy kompilacji i interpretacji kodu,
5. opisać rodzaje prostych typów danych,
6. podać przykłady wykorzystania danych typu prostego,
7. deklarować zmienne prostych typów danych,
8. podać przykłady wyrażen z operatorami arytmetycznymi, przypisania, porównania, logicznymi, operatorami do obsługi łańcuchów, bitowymi oraz wykorzystujących priorytety operatorów,
9. omówić rodzaje złożonych typów danych,
10. identyfikować dane typu złożonego i określić ich rodzaj (tablice jednowymiarowe i dwuwymiarowe, plik, rekord, np. struktura, unia),
11. opisać własności złożonych typów danych
12. identyfikować tablice dynamiczne, asocjacyjne,
13. identyfikować typ wskaźnikowy,
14. zidentyfikować zmienną typu prostego i określić jej rodzaj (numeryczny stałoprzecinkowy i zmiennoprzecinkowy, logiczny, znakowy, łańcuchowy),
15. określić własne typy danych dla problemu programistycznego,
16. podać przykłady wykorzystania złożonych typów danych (tablice, rekordy, pliki, typ wskaźnikowy),
17. objaśnić sposoby przekazywania argumentów funkcji,
18. zapisywać program w postaci zestawu funkcji,
19. identyfikować wybrane biblioteki języka C++, C#, Python lub innego języka programowania: standardowej, z funkcjami matematycznymi, z podstawowymi algorytmami,
20. zdefiniować funkcję rekurencyjną,
21. określić mechanizm obsługi wyjątków z instrukcjami try i catch,
22. zgłaszać wyjątki do obsłużenia (instrukcja throw).

Ocena celująca:

13. Uczeń spełnia wymagania na ocenę bardzo dobrą i samodzielnie poszerza swoją wiedzę z zakresu przedmiotów zawodowych.
14. Uczeń wykazuje zainteresowanie problematyką przedmiotową, a jego zaangażowanie potwierdzone jest aktywnością na zajęciach i poza nimi.
15. Uczeń wykonuje dodatkowe zadania. Pomysłowo i oryginalnie rozwiązuje nietypowe zadania;
16. Uczeń odnosi sukcesy w konkursach, olimpiadach, itp.

Ocena bardzo dobra:

10. Uczeń opanował pełny zakres wiedzy i umiejętności określonych programem nauczania .
11. Uczeń sprawnie posługuje się zdobytymi wiadomościami, potrafi ocenić sytuację, wyciągać prawidłowe wnioski.
12. Uczeń sprawnie wykonuje wszystkie zadania.

Ocena dobra:

10. Uczeń dobrze opanował wiedzę i umiejętności określone programem nauczania.
11. Uczeń dysponuje dobrą umiejętnością zastosowania zdobytych wiadomości.
12. Uczeń uzyskuje stałe dobre postępy podczas zajęć, bierze aktywny udział w lekcji.

Ocena dostateczna:

13. Uczeń opanował wiadomości i umiejętności umożliwiające zdobywanie dalszej wiedzy.
14. Uczeń dysponuje przeciętną wiedzą w zakresie materiału przewidzianego programem nauczania, w jego wiadomościach są luki.
15. Uczeń nie wykazuje większego zainteresowania przedmiotem.
16. Wykonywanie zadań przez ucznia budzi zastrzeżenia.

Ocena dopuszczająca:

13. Zdobyte wiadomości są niewystarczające na uzyskanie przez ucznia podstawowej wiedzy z przedmiotu.
14. Uczeń zadania o niewielkim stopniu trudności rozwiązuje przy pomocy nauczyciela.
15. Uczeń niechętnie bierze udział w lekcjach.
16. Uczeń ma lekceważący stosunek do przedmiotu

Ocena niedostateczna:

7. Uczeń nie opanował wiadomości i umiejętności, jakie były nabywane w danej klasie, braki w wiadomościach i umiejętnościach uniemożliwiają dalsze zdobywanie wiedzy;
8. Uczeń nie potrafi rozwiązywać zadań o niewielkim stopniu trudności.

WYMAGANIA EDUKACYJNE I KRYTERIA OCENIANIA Z PRZEDMIOTU

PROGRAMOWANIE OBIEKTOWE

Kwalifikacja: INF.04 – Technik Programista

Program: Program nauczania dla zawodu Technik Informatyk, 351406, o strukturze przedmiotowej, 2019 r.

Wymiar: 60 godzin w cyklu

Na ocenę dopuszczającą uczeń:

- Zna podstawowe pojęcia związane z programowaniem.
- Zna podstawy algorytmiki
- Potrafi wymienić bloki w schemacie blokowym.
- Odczytuje ze zrozumieniem schematy blokowe.
- Zna programy służące do tworzenia kodu w języku programowania.
- Potrafi wstawić komentarze.
- Potrafi wyjaśnić co to jest zmienna
- Potrafi zadeklarować zmienną
- Wie jakie są typy danych
- Zna operatory w języku C++ oraz wyrażenia
- Zna podstawowe elementy języka C++ takie jak zmienna, typ danych, stała, operator.
- Wie co to jest konwersja typów oraz przestrzeń nazw.
- Zna strukturę programu w języku C++
- Zna składnię instrukcji warunkowej if
- Zna składnię instrukcji switch
- Zna składnię pętli for
- Zna pętlę while i do...while
- Wie jak jest różnica pomiędzy tablicą dynamiczną a statyczną
- Wie co to jest pętla foreach
- Wie jak deklarować funkcje
- Wie jak definiować funkcje
- Wie jak wywołać funkcji w programie.
- Zna różnicę pomiędzy zmiennymi globalnymi i lokalnymi
- Wie co to są dyrektywa #include
- Zna predefiniowane funkcje matematyczne
- Zna funkcje wejścia/wyjścia
- Zna zasady programowania obiektowego
- Wie jak definiować klasy
- Wie jak deklarować zmienne obiektowe w klasie
- Zna zasady odwołania się do elementów członkowskich obiektów
- Wie co to jest destruktork
- Wie co to jest dziedziczenie publiczne (public)
- Wie co to jest dziedziczenie chronione (protected)
- Wie co to jest dziedziczenie prywatne (private)
- Zna mechanizm obsługi wyjątków

Na ocenę dostateczną uczeń spełnia wymagania na ocenę dopuszczającą a dodatkowo:

- Wie co to jest typ wyliczeniowy
- Wie jaka jest różnica pomiędzy statyczną a dynamiczną alokacją pamięci

- Wie co to są tablice statyczne
- Zna różnicę pomiędzy pętlą for i foreach
- Wie co to są struktury w języku C
- Wie co to jest wskaźnik
- Zna różnicę pomiędzy definiowaniem struktury statycznie i dynamicznie
- Wie co to są unie w języku C
- Zna różnicę pomiędzy deklaracją a definicją funkcji
- Wie co to są dyrektywa #define
- Wie co to są funkcje predefiniowane
- Zna predefiniowane funkcje znakowe
- Zna statyczne elementy członkowskie klas
- Wie co to jest konstruktor
- Wie jak zainicjalizować obiekty
- Wie co to jest konstruktor
- Wie jak zainicjalizować obiekty
- Wie co to jest polimorfizm statyczny

Na ocenę dobrą uczeń spełnia wymagania na ocenę dostateczną a dodatkowo:

- Zna pojęcie operatora adresu
- Zna pojęcie operatora deferencji
- Wie co to są łańcuchy typu C-napisy
- Wie co to są łańcuchy string
- Zna funkcje łańcuchowe
- Zna różnicę pomiędzy parametrami funkcji
- Zna różnicę pomiędzy przekazywaniem parametrów funkcji przez wartość i przez referencję
- Rozumie na czym polega funkcja rekurencyjna
- Wie na czym polegają dyrektywy kompilacji warunkowej
- Wie jak dokonać konwersji typu danych
- Wie jak definiować funkcje członkowskie typu inline
- Wie co to jest konstruktor kopiujący
- Wie na czym polega hermetyzacja danych i ukrywanie danych
- Zna zasady definiowania relacji dziedziczenia
- Wie co to jest polimorfizm dynamiczny
- Wie co to są funkcje zaprzyjaźnione
- Zna system komunikatów i kodów zwrotnych

Na ocenę bardzo dobrą uczeń spełnia wymagania na ocenę dobrą a dodatkowo:

- Zna zależności pomiędzy tablicą a wskaźnikiem
- Zna pojęcie wektory
- Wie co to są funkcje przeciążone
- Wie jak wywołać funkcje inline
- Wie co oznacza delegowanie konstruktorów
- Zna rodzaje dziedziczenia
- Wie co to są klasy abstrakcyjne
- Wie co to są interfejsy w klasach abstrakcyjnych
- Zna zasady realizacji mechanizmów abstrakcji w plikach nagłówkowych
- Wie co to są klasy zaprzyjaźnione

Na ocenę celującą uczeń spełnia wymagania na ocenę bardzo dobrą a dodatkowo:

- Wie jak użyć wskaźników jako parametr funkcji
- Zna wiele innych funkcji bibliotecznych
- Wie jak definiować wskaźniki do obiektów
- Wie na czym polega przekazywanie obiektów jako parametrów funkcji
- Zna zależności pomiędzy dziedziczeniem a konstruktorem
- Wie co to jest abstrakcja danych

- Wie co to są szablony funkcji
- Wie na czym polegają szablony klas
- osiągnie sukces w konkursie związanym z tematyką zajęć;
- wykona dodatkowy projekt, wykazując się dodatkowymi umiejętnościami, zgodny z nauczonym przedmiotem. Projekt ma być ustalony z nauczycielem i realizowany systematycznie, a postępy będą prezentowane nauczycielowi w ciągu całego roku szkolnego lub semestru;
- osiągnie średnią ocen zgodną z WSO, klasyfikującą do wystawienia oceny celującej.

Wymagania edukacyjne i kryteria oceniania z przedmiotu

PROGRAMOWANIE OBIEKTOWE

Kwalifikacja: INF.04 – Technik Programista

Program: Program nauczania dla zawodu Technik Informatyk, 351406, o strukturze przedmiotowej, 2019 r.

Wymiar: 60 godzin w cyklu

TABELA

Nr	Tematy	Oceny				
		2	3	4	5	6
1	Zna podstawowe pojęcia związane z programowaniem.	x	x	x	x	x
2	Zna podstawy algorytmiki	x	x	x	x	x
3	Potrafi wymienić bloki w schemacie blokowym.	x	x	x	x	x
4	Odczytuje ze zrozumieniem schematy blokowe.	x	x	x	x	x
5	Zna programy służące do tworzenia kodu w języku programowania.	x	x	x	x	x
6	Potrafi wstawić komentarze.	x	x	x	x	x
7	Potrafi wyjaśnić co to jest zmienna	x	x	x	x	x
8	Potrafi zadeklarować zmienną	x	x	x	x	x
9	Wie jakie są typy danych	x	x	x	x	x
10	Zna operatory w języku C++ oraz wyrażenia	x	x	x	x	x
11	Zna podstawowe elementy języka C++ takie jak zmienna, typ danych, stała, operator.	x	x	x	x	x
12	Wie co to jest konwersja typów oraz przestrzeń nazw.	x	x	x	x	x
13	Zna strukturę programu w języku C++	x	x	x	x	x
14	Zna składnię instrukcji warunkowej if	x	x	x	x	x
15	Zna składnię instrukcji switch	x	x	x	x	x
16	Zna składnię pętli for	x	x	x	x	x
17	Zna pętlę while i do...while	x	x	x	x	x
18	Wie co to jest typ wyliczeniowy		x	x	x	x
19	Wie co to jest wskaźnik		x	x	x	x
20	Zna pojęcie operatora adresu			x	x	x
21	Zna pojęcie operatora deferencji			x	x	x
22	Wie jaka jest różnica pomiędzy statyczną a dynamiczną alokacją pamięci		x	x	x	x
23	Wie co to są tablice statyczne		x	x	x	x
24	Zna zależności pomiędzy tablicą a wskaźnikiem				x	x
25	Wie jak jest różnica pomiędzy tablicą dynamiczną a statyczną	x	x	x	x	x
26	Zna pojęcie wektory				x	x

27	Wie co to jest pętla foreach	x	x	x	x	x
28	Zna różnicę pomiędzy pętlą for i foreach		x	x	x	x
29	Wie co to są łańcuchy typu C-napisy			x	x	x
30	Wie co to są łańcuchy string			x	x	x
31	Zna funkcje łańcuchowe			x	x	x
32	Wie co to są struktury w języku C		x	x	x	x
33	Zna różnicę pomiędzy definiowaniem struktury statycznie i dynamicznie		x	x	x	x
34	Wie co to są unie w języku C		x	x	x	x
35	Zna różnicę pomiędzy deklaracją a definicją funkcji		x	x	x	x
36	Wie jak deklarować funkcje	x	x	x	x	x
37	Wie jak definiować funkcje	x	x	x	x	x
38	Wie jak wywołać funkcji w programie.	x	x	x	x	x
39	Zna różnicę pomiędzy parametrami funkcji			x	x	x
40	Zna różnicę pomiędzy przekazywaniem parametrów funkcji przez wartość i przez referencję			x	x	x
41	Wie jak użyć wskaźników jako parametr funkcji.					x
42	Zna różnicę pomiędzy zmiennymi globalnymi i lokalnymi.	x	x	x	x	x
43	Wie co to są funkcje przeciążone				x	x
44	Wie jak wywołać funkcje inline				x	x
45	Rozumie na czym polega funkcja rekurencyjna			x	x	x
46	Wie co to są dyrektywa #include	x	x	x	x	x
47	Wie co to są dyrektywa #define		x	x	x	x
48	Wiena czym polegają dyrektywy kompilacji warunkowej			x	x	x
49	Wie co to są funkcje predefiniowane		x	x	x	x
50	Zna predefiniowane funkcje matematyczne	x	x	x	x	x
51	Zna predefiniowane funkcje znakowe		x	x	x	x
52	Wie jak dokonać konwersji typu danych			x	x	x
53	Zna funkcje wejścia/wyjścia	x	x	x	x	x
54	Zna wiele innych funkcji bibliotecznych					x
55	Zna zasady programowania obiektowego	x	x	x	x	x
56	Wie jak definiować klasy	x	x	x	x	x
57	Wie jak deklarować zmienne obiektowe w klasie	x	x	x	x	x
58	Zna zasady odwołania się do elementów członkowskich obiektów	x	x	x	x	x
59	Zna statyczne elementy członkowskie klas		x	x	x	x
60	Wie jak definiować funkcje członkowskie typu inline			x	x	x
61	Wie jak definiować wskaźniki do obiektów					x
62	Wie na czym polega przekazywanie obiektów jako parametrów funkcji					x
63	Wie co to jest konstruktor		x	x	x	x
64	Wie jak zainicjalizować obiekty		x	x	x	x
65	Wie co to jest konstruktor kopiujący			x	x	x
66	Wie co oznacza delegowanie konstruktorów				x	x

67	Wie co to jest destruktork	x	x	x	x	x
68	Wie na czym polega hermetyzacja danych i ukrywanie danych			x	x	x
69	Zna zasady definiowania relacji dziedziczenia			x	x	x
70	Wie co to jest dziedziczenie publiczne (public)	x	x	x	x	x
71	Wie co to jest dziedziczenie chronione (protected)	x	x	x	x	x
72	Wie co to jest dziedziczenie prywatne (private)	x	x	x	x	x
73	Zna rodzaje dziedziczenia				x	x
74	Zna zależności pomiędzy dziedziczeniem a konstruktorem					x
75	Wie co to jest polimorfizm statyczny		x	x	x	x
76	Wie co to jest polimorfizm dynamiczny			x	x	x
77	Wie co to są klasy abstrakcyjne				x	x
78	Wie co to są interfejsy w klasach abstrakcyjnych				x	x
79	Wie co to jest abstrakcja danych					x
80	Zna zasady realizacji mechanizmów abstrakcji w plikach nagłówkowych				x	x
81	Wie co to są funkcje zaprzyjaźnione			x	x	x
82	Wie co to są klasy zaprzyjaźnione				x	x
83	Wie co to są szablony funkcji					x
84	Wie na czym polegają szablony klas					x
85	Zna system komunikatów i kodów zwrotnych			x	x	x
86	Zna mechanizm obsługi wyjątków	x	x	x	x	x

WYMAGANIA EDUKACYJNE I KRYTERIA OCENIANIA Z PRZEDMIOTU PRACOWNIA PROGRAMOWANIA OBIEKTOWEGO

Kwalifikacja: INF.04 – Technik Programista

Program: Program nauczania dla zawodu Technik Informatyk, 351406, o strukturze przedmiotowej, 2019 r.

Wymiar: 90 godzin w cyklu

Na ocenę dopuszczającą uczeń:

- Wykonuje algorytmy różnego rodzaju (pseudokod, opis słowny)
- Tworzy algorytmy za pomocą schematu blokowego.
- Posługuje się programami za pomocą których tworzy i kompiluje kodu napisany w języku programowania.
- Deklaruje przestrzeń nazw w języku C++
- Wstawia komentarze w kodzie programu.
- Potrafi zadeklarować zmienną
- Używa w kodzie programy różnego typu zmiennych
- Stosuje operatory w języku C++
- Dokonuje konwersji typów w C++
- Stosuje odpowiednią strukturę programu w języku C++
- Wykorzystuje składnię instrukcji warunkowej if
- Stosuje w kodzie programu składnię instrukcji switch
- Stosuje w kodzie programu składnię pętli for
- Stosuje w kodzie programu pętlę while i do...while
- Deklaruje tablice statyczne
- Odwołuje się do elementów składowych tablic
- Stosuje pętlę foreach
- Tworzy łańcuchy typu string w C++
- Deklaruje zmienne strukturalne
- Potrafi odwołać się do pól struktury
- Potrafi zadeklarować funkcję
- Potrafi napisać definicję funkcji
- Wywołuje funkcji w programie
- Deklaruje zmienne globalne i lokalne
- Deklaruje dyrektywę #include
- Deklaruje dyrektywę #define
- Stosuje funkcje predefiniowane
- Stosuje predefiniowane funkcje matematyczne
- Używa funkcji wejścia/wyjścia
- Potrafi zdefiniować klasy
- Wie jak deklarować zmienne obiektowe w klasie
- Potrafi deklarować zmienne obiektowe
- Potrafi odwołać się do elementów członkowskich obiektów
- Potrafi zdefiniować destruktor
- Stosuje hermetyzację danych (ukrywanie danych)
- Używa publicznych specyfikatorów dostępu (public)

Na ocenę dostateczną uczeń spełnia wymagania na ocenę dopuszczającą a dodatkowo:

- Definiuje typy wyliczeniowe w języku C++
- Deklaruje zmienne wyliczeniowe w C++

- Stosuje wskaźnik w programie.
- Wykorzystuje łańcuchy typu C-napisy w programie C++
- Stosuje funkcje łańcuchowe
- Deklaruje i definiuje struktury w języku C
- Tworzy struktury dynamiczne
- Definiuje unie w języku C
- Potrafi przekazać parametry funkcji przez wartość
- Potrafi przekazać parametry funkcji przez referencję
- Stosuje predefiniowane funkcje znakowe
- Deklaruje statyczne elementy członkowskie klas
- Potrafi definiować funkcje członkowskie (metody)
- Potrafi zdefiniować konstruktor
- Potrafi zainicjalizować obiekty
- Potrafi zainicjalizować obiekty
- Używa chronionych specyfikatorów dostępu (protected)
- Używa prywatnych specyfikatorów dostępu (private)
- Stosuje mechanizm przeciążenia metod w polimorfizmie statycznym
- Stosuje mechanizm przesłania metod w polimorfizmie statycznym
- Stosuje mechanizm obsługi wyjątków

Na ocenę dobrą uczeń spełnia wymagania na ocenę dostateczną a dodatkowo:

- Stosuje operator adresu
- stosuje operator deferencji
- Inicjalizuje i odwołuje się do zmiennych typu unijnego
- Świadomie różnicuje przekazywanie parametrów funkcji przez wartość i przez referencję
- Tworzy funkcja rekurencyjna
- Dokonuje konwersji typu danych
- Potrafi definiować funkcje członkowskie typu inline
- Potrafi dokonać przeciążenia konstruktorów
- Potrafi zdefiniować konstruktor kopiujący
- Potrafi zdefiniować relacje dziedziczenia
- Stosuje mechanizm polimorfizmu dynamicznego
- Stosuje funkcje zaprzyjaźnione
- Potrafi zastosować system komunikatów i kodów zwrotnych

Na ocenę bardzo dobrą uczeń spełnia wymagania na ocenę dobrą a dodatkowo:

- Stosuje operatory bitowe w języku C++
- Przydziela wskaźnikom pamięć operacyjną w sposób dynamiczny za pomocą operatora new
- Zwalnia pamięć operacyjną po dynamicznym przydzieleniu za pomocą operatora delete
- Deklaruje tablice wielowymiarowe
- Odwołuje się do elementów składowych tablic wielowymiarowych
- Deklaruje tablice dynamiczne za pomocą operatora New
- Definiuje funkcje przeciążone
- Wywołuje funkcje inline
- Stosuje dyrektywy kompilacji warunkowej
- Potrafi zdefiniować delegowanie konstruktorów
- Tworzy klasy abstrakcyjne
- Tworzy klasy zaprzyjaźnione

Na ocenę celującą uczeń spełnia wymagania na ocenę bardzo dobrą a dodatkowo:

- Deklaruje wskaźniki wskazujące na tablice
- Stosuje predefiniowaną funkcję vector do przechowywania list określonych danych
- Wykorzystuje wskaźniki do struktur
- Używa wskaźników jako parametr funkcji
- Wykorzystuje wiele innych funkcji bibliotecznych

- Deklaruje wskaźniki do obiektów
- Potrafi przekazać obiekty jako parametrów funkcji
- Tworzy metody wirtualne w poliformizmie dynamicznym
- Tworzy dane abstrakcyjne
- Stosuje szablony funkcji i klas
- osiągnie średnią ocen zgodną z WSO, klasyfikującą do wystawienia oceny celującej.

Wymagania edukacyjne i kryteria oceniania z przedmiotu
PRACOWNIA PROGRAMOWANIA OBIEKTOWEGO

Kwalifikacja: INF.04 – Technik Programista

Program: Program nauczania dla zawodu Technik Informatyk, 351406, o strukturze przedmiotowej, 2019 r.

Wymiar: 90 godzin w cyklu

TABELA

Nr	Zagadnienia	Oceny				
		2	3	4	5	6
1	Wykonuje algorytmy różnego rodzaju (pseudokod, opis słowny)	x	x	x	x	x
2	Tworzy algorytmy za pomocą schematu blokowego.	x	x	x	x	x
3	Posługuje się programami za pomocą których tworzy i kompiluje kodu napisany w języku programowania.	x	x	x	x	x
4	Deklaruje przestrzeń nazw w języku C++	x	x	x	x	x
5	Wstawia komentarze w kodzie programu.	x	x	x	x	x
6	Potrafi zadeklarować zmienną	x	x	x	x	x
7	Używa w kodzie programy różnego typu zmiennych	x	x	x	x	x
8	Stosuje operatory w języku C++	x	x	x	x	x
9	Stosuje operatory bitowe w języku C++				x	x
10	Dokonuje konwersji typów w C++	x	x	x	x	x
11	Stosuje odpowiednią strukturę programu w języku C++	x	x	x	x	x
12	Wykorzystuje składnię instrukcji warunkowej if	x	x	x	x	x
13	Stosuje w kodzie programu składnię instrukcji switch	x	x	x	x	x
14	Stosuje w kodzie programu składnię pętli for	x	x	x	x	x
15	Stosuje w kodzie programu pętlę while i do...while	x	x	x	x	x
16	Definiuje typy wyliczeniowe w języku C++		x	x	x	x
17	Deklaruje zmienne wyliczeniowe w C++		x	x	x	x
18	Stosuje wskaźnik w programie.		x	x	x	x
19	Stosuje operator adresu			x	x	x
20	stosuje operator deferencji			x	x	x
21	Przydziela wskaźnikom pamięć operacyjną w sposób dynamiczny za pomocą operatora new				x	x
22	Zwalnia pamięć operacyjną po				x	x

	dynamicznym przydzieleniu za pomocą operatora delete					
23	Deklaruje tablice statyczne	x	x	x	x	x
24	Odwołuje się do elementów składowych tablic	x	x	x	x	x
25	Deklaruje tablice wielowymiarowe				x	x
26	Odwołuje się do elementów składowych tablic wielowymiarowych				x	x
27	Deklaruje wskaźniki wskazujące na tablice					x
28	Deklaruje tablice dynamiczne za pomocą operatora new				x	x
29	Stosuje predefiniowaną funkcję vector do przechowywania list określonych danych					x
30	Stosuje pętlę foreach	x	x	x	x	x
31	Wykorzystuje łańcuchy typu C-napisy w programie C++		x	x	x	x
32	Tworzy łańcuchy typu string w C++	x	x	x	x	x
33	Stosuje funkcje łańcuchowe		x	x	x	x
34	Deklaruje i definiuje struktury w języku C		x	x	x	x
35	Deklaruje zmienne strukturalne	x	x	x	x	x
36	Potrafi odwołać się do pól struktury	x	x	x	x	x
37	Wykorzystuje wskaźniki do struktur					x
38	Tworzy struktury dynamiczne		x	x	x	x
39	Definiuje unie w języku C		x	x	x	x
40	Inicjalizuje i odwołuje się do zmiennych typu unijnego			x	x	x
41	Potrafi zadeklarować funkcję	x	x	x	x	x
42	Potrafi napisać definicję funkcji	x	x	x	x	x
43	Wywołuje funkcji w programie.	x	x	x	x	x
44	Potrafi przekazać parametry funkcji przez wartość		x	x	x	x
45	Potrafi przekazać parametry funkcji przez referencję		x	x	x	x
46	Świadomie różnicuje przekazywanie parametrów funkcji przez wartość i przez referencję.			x	x	x
47	Używa wskaźników jako parametr funkcji.					x
48	Deklaruje zmienne globalne i lokalne.	x	x	x	x	x
49	Definiuje funkcje przeciążone				x	x
50	Wywołuje funkcje inline				x	x
51	Tworzy funkcja rekurencyjna			x	x	x
52	Deklaruje dyrektywę #include	x	x	x	x	x
53	Deklaruje dyrektywę #define	x	x	x	x	x
54	Stosuje dyrektywy kompilacji warunkowej				x	x
55	Stosuje funkcje predefiniowane	x	x	x	x	x
56	Stosuje predefiniowane funkcje matematyczne	x	x	x	x	x
57	Stosuje predefiniowane funkcje znakowe		x	x	x	x
58	Dokonuje konwersji typu danych			x	x	x
59	Używa funkcji wejścia/wyjścia	x	x	x	x	x

60	Wykorzystuje wiele innych funkcji bibliotecznych					X
61	Potrafi zdefiniować klasy	X	X	X	X	X
62	Wie jak deklorować zmienne obiektowe w klasie	X	X	X	X	X
63	Potrafi deklorować zmienne obiektowe	X	X	X	X	X
64	Potrafi odwołać się do elementów członkowskich obiektów	X	X	X	X	X
65	Deklaruje statyczne elementy członkowskie klas		X	X	X	X
66	Potrafi definiować funkcje członkowskie (metody)		X	X	X	X
67	Potrafi definiować funkcje członkowskie typu inline			X	X	X
68	Deklaruje wskaźniki do obiektów					X
69	Potrafi przekazać obiekty jako parametrów funkcji					X
70	Potrafi zdefiniować konstruktor		X	X	X	X
71	Potrafi dokonać przeciążenia konstruktorów			X	X	X
72	Potrafi zainicjalizować obiekty		X	X	X	X
73	Potrafi zdefiniować konstruktor kopiujący			X	X	X
74	Potrafi zdefiniować delegowanie konstruktorów				X	X
75	Potrafi zdefiniować destruktory	X	X	X	X	X
76	Stosuje hermetyzację danych (ukrywanie danych)	X	X	X	X	X
77	Potrafi zdefiniować relacje dziedziczenia			X	X	X
78	Używa publicznych specyfikatorów dostępu (public)	X	X	X	X	X
79	Używa chronionych specyfikatorów dostępu (protected)		X	X	X	X
80	Używa prywatnych specyfikatorów dostępu (private)		X	X	X	X
81	Stosuje mechanizm przeciążenia metod w polimorfizmie statycznym		X	X	X	X
82	Stosuje mechanizm przesłania metod w polimorfizmie statycznym		X	X	X	X
83	Stosuje mechanizm polimorfizmu dynamicznego			X	X	X
84	Tworzy metody wirtualne w polimorfizmie dynamicznym					X
85	Tworzy klasy abstrakcyjne				X	X
86	Tworzy dane abstrakcyjne					X
87	Stosuje funkcje zaprzyjaźnione			X	X	X
88	Tworzy klasy zaprzyjaźnione				X	X
89	Stosuje szablony funkcji i klas					X
90	Potrafi zastosować system komunikatów i kodów zwrotnych			X	X	X
91.	Stosuje mechanizm obsługi wyjątków		X	X	X	X

PRZEDMIOT: ROZWÓJ KOMPETENCJI ZAWODOWYCH

kwalifikacja inf.03 i inf.04

Efekty kształcenia:

- Uczeń stosuje podstawowe elementy języków skryptowych
- Potrafi zastosować poszczególne znaczniki
- Potrafi tworzyć proste strony oparte na znacznikach sekcji
- Tworzy nagłówki i paragrafy
- Stosuje elementy list, tabel, obrazów, odsyłaczy, formularzy z podstawowymi elementami
- Zapisuje zdarzenia i skrypty
- Tworzy style wewnętrzne i zewnętrzne
- Wykorzystuje kaskadowość stylów CSS
- Wykorzystuje różnego rodzaju selektory
- Stosuje reguły CSS (składnia stylu wewnętrznego zewnętrznego i inline)
- Tworzy różne projekty graficzne strony internetowej
- Przypisuje barwy dla różnych elementów na stronie internetowej
- Tworzy fragmenty tekstu z różną wielkością czcionki na przykładowych stronach internetowych
- Testuje poprawność strony internetowej
- Waliduje wcześniej stworzone strony internetowe
- Pozycjonuje strony WWW
- Uczeń stosuje przykłady języków skryptowych
- Potrafi zastosować poszczególne znaczniki
- Potrafi tworzyć proste strony oparte na znacznikach sekcji
- Tworzy nagłówki i paragrafy
- Stosuje elementy list, tabel, obrazów, odsyłaczy, formularzy
- Zapisuje zdarzenia i skrypty
- Tworzy proste formularze wraz z polami w standardzie HTML5
- Tworzy style wewnętrzne i zewnętrzne
- Wykorzystuje kaskadowość stylów CSS
- Wykorzystuje różnego rodzaju selektory
- Stosuje reguły CSS (składnia stylu wewnętrznego zewnętrznego i inline)
- Tworzy różne projekty graficzne strony internetowej
- Przypisuje barwy dla różnych elementów na stronie internetowej

- Tworzy fragmenty tekstu z różną wielkością czcionki na przykładowych stronach internetowych
- Testuje poprawność strony internetowej
- Waliduje wcześniej stworzone strony internetowe
- Pozycjonuje strony WWW
- Uczeń stosuje przykłady języków skryptowych
- Potrafi zastosować poszczególne znaczniki
- Potrafi tworzyć proste strony oparte na znacznikach sekcji
- Tworzy nagłówki i paragrafy
- Stosuje elementy list, tabel, obrazów, odsyłaczy, formularzy
- Zapisuje zdarzenia i skrypty
- Tworzy formularze wraz z polami w standardzie HTML5
- Tworzy style wewnętrzne i zewnętrzne
- Wykorzystuje kaskadowość stylów CSS
- Wykorzystuje różnego rodzaju selektory
- Stosuje reguły CSS (składnia stylu wewnętrznego zewnętrznego i inline)
- Tworzy różne projekty graficzne strony internetowej
- Przypisuje barwy dla różnych elementów na stronie internetowej
- Tworzy fragmenty tekstu z różną wielkością czcionki na przykładowych stronach internetowych
- Waliduje wcześniej stworzone strony internetowe
- Pozycjonuje strony WWW
- Uczeń stosuje zaawansowane przykłady języków skryptowych
- Potrafi zastosować wszystkie znaczniki HTML-a
- Potrafi tworzyć zaawansowane strony oparte na znacznikach sekcji
- Tworzy nagłówki i paragrafy
- Definiuje elementy list, tabel, obrazów, odsyłaczy, formularzy, multimediiów
- Zapisuje zdarzenia i rozbudowane skrypty
- Tworzy formularze wraz z polami w standardzie HTML5
- Tworzy style wewnętrzne i zewnętrzne
- Wykorzystuje kaskadowość stylów CSS
- Wykorzystuje różnego rodzaju selektory
- Stosuje reguły CSS (składnia stylu wewnętrznego zewnętrznego i inline)
- Tworzy szablony stron wraz z stylem CSS

- Tworzy responsywny układ strony
- Tworzy różne projekty graficzne strony internetowej
- Tworzy fragmenty tekstu z różną wielkością czcionki na przykładowych stronach internetowych
- Testuje poprawność strony internetowej
- Waliduje wcześniej stworzone strony internetowe
- Pozycjonuje strony WWW
- Sprawdza czy strona spełnia wymagania w zakresie optymalizacji
- Wykorzystuje narzędzia do testowania szybkości wczytywania strony internetowej
- Publikuje strony internetowe
- Wykorzystuje program FileZilla
- charakteryzuje podstawowe pojęcia: aplikacja internetowa, skrypt, aplet
- potrafi zastosować algorytm do opisu problemu programistycznego
- stosuje narzędzia programistyczne
- potrafi zastosować język JavaScript do tworzenia aplikacji internetowych
- stosuje język PHP do tworzenia aplikacji internetowych
- posługuje się typami prostymi i złożonymi oraz zmiennymi przy tworzeniu aplikacji w językach JavaScript, PHP oraz C#
- posługuje się operatorami przy tworzeniu aplikacji w językach JavaScript, PHP oraz C#
- potrafi zastosować instrukcje sterujące języków skryptowych
- potrafi zastosować funkcje oraz wybrane biblioteki języków skryptowych
- tworzy strony internetowe wykorzystujące skryptowe języki programowania
- stosuje funkcje języka JavaScript
- stosuje skrypty obsługujące formularze i kontrolki HTML
- stosuje skrypt PHP w kodzie HTML
- potrafi zastosować funkcje języka PHP
- tworzy skrypty przekazujące dane z formularza
- definiuje połączenie z bazą danych
- definiuje zapytania do bazy danych
- tworzy programy w wybranym środowisku programistycznym (XAMPP, Visual Studio)
- instaluje i konfiguruje serwer WWW (Apache, ASP.NET)
- instaluje i konfiguruje serwer MS SQL
- instaluje i konfiguruje serwer MySQL
- korzysta z pakietu phpMyAdmin

- analizuje błędy w kodzie źródłowym programu
- poprawia błędy w tworzonych programach
- stosuje debugger w przeglądarce Internetowej
- potrafi zastosować komentarze w kodzie źródłowym programu
- potrafi zastosować algorytm do opisu problemu programistycznego
- stosuje narzędzia programistyczne
- potrafi zastosować język JavaScript do tworzenia aplikacji internetowych
- stosuje język PHP do tworzenia aplikacji internetowych
- posługuje się typami prostymi i złożonymi oraz zmiennymi przy tworzeniu aplikacji w językach JavaScript, PHP oraz C#
- posługuje się operatorami przy tworzeniu aplikacji w językach JavaScript, PHP oraz C#
- potrafi zastosować instrukcje sterujące języków skryptowych
- potrafi zastosować funkcje oraz wybrane biblioteki języków skryptowych
- tworzy strony internetowe wykorzystujące skryptowe języki programowania
- stosuje funkcje języka JavaScript
- potrafi zastosować obsługę zdarzeń myszy i klawiatury
- stosuje skrypty obsługujące formularze i kontrolki HTML
- stosuje walidację formularzy z wykorzystaniem biblioteki jQuery
- stosuje skrypt PHP w kodzie HTML
- potrafi zastosować funkcje języka PHP
- tworzy skrypty przekazujące dane z formularza
- tworzy skrypty wykorzystujące mechanizm sesji
- definiuje połączenie z bazą danych
- definiuje zapytania do bazy danych
- dobiera środowisko programistyczne do określonych zadań i języka programowania
- tworzy programy w wybranym środowisku programistycznym (XAMPP, Visual Studio)
- instaluje i konfiguruje serwer WWW (Apache, ASP.NET)
- instaluje i konfiguruje serwer MS SQL
- instaluje i konfiguruje serwer MySQL
- korzysta z pakietu phpMyAdmin
- korzysta z pakietu Management Studio
- analizuje błędy w kodzie źródłowym programu
- poprawia błędy w tworzonych programach
- stosuje debugger w przeglądarce Internetowej
- potrafi zastosować komentarze w kodzie źródłowym programu

- tworzy dokumentację programu
- potrafi zastosować algorytm do opisu problemu programistycznego
- stosuje narzędzia programistyczne
- potrafi zastosować język JavaScript do tworzenia aplikacji internetowych
- stosuje język PHP do tworzenia aplikacji internetowych
- posługuje się typami prostymi i złożonymi oraz zmiennymi przy tworzeniu aplikacji w językach JavaScript i PHP
- posługuje się operatorami przy tworzeniu aplikacji w językach JavaScript i PHP
- potrafi zastosować instrukcje sterujące języków skryptowych
- potrafi zastosować funkcje oraz wybrane biblioteki języków skryptowych
- tworzy strony internetowe wykorzystujące skryptowe języki programowania
- stosuje funkcje języka JavaScript
- stosuje obiekty przeglądarki
- potrafi zastosować obiekty dokumentu DOM
- potrafi zastosować obsługę zdarzeń myszy i klawiatury
- stosuje skrypty obsługujące formularze i kontrolki HTML
- stosuje obsługę zdarzeń myszy i klawiatury za pomocą biblioteki jQuery
- stosuje walidację formularzy z wykorzystaniem biblioteki jQuery
- stosuje skrypt PHP w kodzie HTML
- potrafi zastosować funkcje języka PHP
- stosuje funkcje obsługi plików
- tworzy skrypty przekazujące dane z formularza
- tworzy pliki cookies
- tworzy skrypty wykorzystujące mechanizm sesji
- definiuje połączenie z bazą danych
- definiuje zapytania do bazy danych
- dobiera środowisko programistyczne do określonych zadań i języka programowania
- tworzy programy w wybranym środowisku programistycznym (XAMPP, Visual Studio)
- instaluje i konfiguruje serwer WWW (Apache, ASP.NET)
- instaluje i konfiguruje serwer MS SQL
- instaluje i konfiguruje serwer MySQL
- korzysta z pakietu phpMyAdmin
- korzysta z pakietu Management Studio
- analizuje błędy w kodzie źródłowym programu
- wykonuje testy tworzonych programów

- poprawia błędy w tworzonych programach
- stosuje debugger w przeglądarce Internetowej
- potrafi zastosować komentarze w kodzie źródłowym programu
- tworzy dokumentację programu
- tworzy instrukcję użytkownika programu
- potrafi zastosować algorytm do opisu problemu programistycznego
- stosuje narzędzia programistyczne
- potrafi zastosować język JavaScript do tworzenia aplikacji internetowych
- stosuje język PHP do tworzenia aplikacji internetowych
- stosuje język C# do tworzenia aplikacji internetowych
- posługuje się typami prostymi i złożonymi oraz zmiennymi przy tworzeniu aplikacji w językach JavaScript, PHP
- posługuje się operatorami przy tworzeniu aplikacji w językach JavaScript, PHP
- potrafi zastosować instrukcje sterujące języków skryptowych
- potrafi zastosować funkcje oraz wybrane biblioteki języków skryptowych
- tworzy strony internetowe wykorzystujące skryptowe języki programowania
- stosuje funkcje języka JavaScript
- tworzyć obiekty w języku JavaScript
- stosuje obiekty przeglądarki
- stosuje obiekty języka JavaScript
- potrafi zastosować obiekty dokumentu DOM
- potrafi zastosować obsługę zdarzeń myszy i klawiatury
- stosuje skrypty obsługujące formularze i kontrolki HTML
- potrafi zastosować wyrażenia regularne do walidacji formularza
- stosuje obsługę zdarzeń myszy i klawiatury za pomocą biblioteki jQuery
- stosuje walidację formularzy z wykorzystaniem biblioteki jQuery
- tworzy skrypty z wykorzystaniem biblioteki AngularJS
- tworzyć skrypty z wykorzystaniem biblioteki React
- stosuje skrypt PHP w kodzie HTML
- potrafi zastosować funkcje języka PHP
- stosuje funkcje obsługi plików
- tworzy skrypty przekazujące dane z formularza
- tworzy pliki cookies
- tworzy skrypty wykorzystujące mechanizm sesji
- definiuje połączenie z bazą danych

Wymagania edukacyjne dla zawodu technik programista 351406

- definiuje zapytania do bazy danych
- potrafi zastosować bibliotekę PDO do obsługi bazy Danych
- dobiera środowisko programistyczne do określonych zadań i języka programowania
- tworzy programy w wybranym środowisku programistycznym (XAMPP, Visual Studio)
- instaluje i konfiguruje serwer WWW (Apache, ASP.NET)
- instaluje i konfiguruje serwer MS SQL
- instaluje i konfiguruje serwer MySQL
- korzysta z pakietu phpMyAdmin
- korzysta z pakietu Management Studio
- analizuje błędy w kodzie źródłowym programu
- wykonuje testy tworzonych programów
- poprawia błędy w tworzonych programach
- stosuje debugger w przeglądarce Internetowej
- potrafi zastosować komentarze w kodzie źródłowym programu
- tworzy dokumentację programu
- tworzy instrukcję użytkownika programu

Ocena celująca:

1. Uczeń spełnia wymagania na ocenę bardzo dobrą i samodzielnie poszerza swoją wiedzę z zakresu przedmiotów zawodowych.
2. Uczeń wykazuje zainteresowanie problematyką przedmiotową, a jego zaangażowanie potwierdzone jest aktywnością na zajęciach i poza nimi.
3. Uczeń wykonuje dodatkowe zadania. Pomysłowo i oryginalnie rozwiązuje nietypowe zadania;
4. Uczeń odnosi sukcesy w konkursach, olimpiadach, itp.

Ocena bardzo dobra:

1. Uczeń opanował pełny zakres wiedzy i umiejętności określonych programem nauczania .
2. Uczeń sprawnie posługuje się zdobytymi wiadomościami, potrafi ocenić sytuację, wyciągać prawidłowe wnioski.
3. Uczeń sprawnie wykonuje wszystkie zadania.

Ocena dobra:

1. Uczeń dobrze opanował wiedzę i umiejętności określone programem nauczania.
2. Uczeń dysponuje dobrą umiejętnością zastosowania zdobytych wiadomości.
3. Uczeń uzyskuje stałe dobre postępy podczas zajęć, bierze aktywny udział w lekcji.

Ocena dostateczna:

1. Uczeń opanował wiadomości i umiejętności umożliwiające zdobywanie dalszej wiedzy.

Wymagania edukacyjne dla zawodu technik programista 351406

2. Uczeń dysponuje przeciętną wiedzą w zakresie materiału przewidzianego programem nauczania, w jego wiadomościach są luki.
3. Uczeń nie wykazuje większego zainteresowania przedmiotem.
4. Wykonywanie zadań przez ucznia budzi zastrzeżenia.

Ocena dopuszczająca:

1. Zdobyte wiadomości są niewystarczające na uzyskanie przez ucznia podstawowej wiedzy z przedmiotu.
2. Uczeń zadania o niewielkim stopniu trudności rozwiązuje przy pomocy nauczyciela.
3. Uczeń niechętnie bierze udział w lekcjach.
4. Uczeń ma lekceważący stosunek do przedmiotu

Ocena niedostateczna:

1. Uczeń nie opanował wiadomości i umiejętności, jakie były nabywane w danej klasie, braki w wiadomościach i umiejętnościach uniemożliwiają dalsze zdobywanie wiedzy;
2. Uczeń nie potrafi rozwiązywać zadań o niewielkim stopniu trudności.