Wymagania edukacyjne z przedmiotu

Aplikacje Desktopowe

Cele ogólne przedmiotu

1. Poznanie

- zagrożeń w środowisku pracy i wymogów w zakresie organizacji stanowiska pracy,
- zasad udzielania pierwszej pomocy,
- algorytmów,
- środowisk programistycznych,
- struktur danych,
- paradygmatów programowania strukturalnego i obiektowego,
- zasady tworzenia okienkowych aplikacji desktopowych,
- zasady testowania i dokumentowania aplikacji.

2. Nabycie umiejętności

- bezpiecznego wykonywania zadań zawodowych,
- stosowania w programowaniu obiektowości,
- samodzielnego tworzenia poprawnych programów desktopowych,
- wyboru algorytmu do rozwiazywania zadań,
- tworzenia programów według dostarczonego projektu,
- dokumentowania programu.

3. Kształtowanie postaw i świadomości

- Aktywnego słuchania i prowadzenia dyskusji;
- Stosowania zasad kultury osobistej i ogólnie przyjętych norm zachowania w swoim środowisku,
- Wrażliwości na potrzeby osób niepełnosprawnych,
- Odpowiedzialności za podejmowane działania,
- Kreatywności w rozwiazywaniu problemów,
- Stosowania właściwej techniki twórczego myślenia przy rozwiązywaniu problemów,
- Doskonalenia jakości wykonywanych działań,
- Analizowania i oceny podejmowanych działań,
- Doskonalenia umiejętności zawodowych,
- Pracy w zespole.

Efekty kształcenia uczeń

1. Tworzy użyteczny kod programistyczny.
2. Rozróżnia typy danych w różnych językach.
3. Rozumie zasadę działania typów numerycznych.
4. Rozumie działanie typów logicznych.
5. Potrafi wykorzystać flagę w kodzie.
6. Rozróżnia ASCII od UNICODE.
7. Potrafi scharakteryzować najważniejsze typy danych.
8. Potrafi zaprogramować tablice jednowymiarowe i dwuwymiarowe.
9. Różnicuje tablice dynamiczne i asocjacyjne

10. Potrafi opisać typ struct i unia.
11. Potrafi wyjaśnić budowę i funkcjonowanie wskaźników w C++
12. Potrafi operować na typach plikowych.
13. Rozróżnia kilka rodzajów kolekcji.
14. Rozumie podstawowe algorytmy programistyczne.
15. Potrafi opisać algorytmy iteracyjne tekstowe i rekurencyjne.
16. Potrafi wyjaśnić różnicę między tradycyjnym podejściem do planowania i zarządzania projektem a filozofią Agile.
17. Rozumie zasady wizualizacji etapów realizacji projektu.
18. Potrafi opisać oprogramowanie wspierające zarządzanie projektem.
19. Rozumie zasady funkcjonowania zespołu projektowego.
20. Rozumie algorytmy sortowania i wyszukiwania.
21. Rozumie podstawowe wzorce projektowe.
22. Potrafi rozróżniać autorskie prawa osobiste i majątkowe.
23. Potrafi opisać środowisko programistyczne dla określonego zadania.
24. Potrafi zastosować podstawowe zasady programowania w tym rekurencję.
25. Potrafi zastosować proste złożone typy danych.
26. Potrafi użyć instrukcje i biblioteki
27. Rozumie podstawowe pojęcia programowania obiektowego.
28. Rozumie podstawy teoretyczne klas w programowaniu.
29. Potrafi tworzyć obiekty na bazie klas.
30. Potrafi zdefiniować zaawansowane zagadnienia związane z klasami w programowaniu.
31. Rozumie podstawowe zasady przechwytywania błędów.
32. Rozróżnia aplikacje typu RAD, IDE proste edytory kod.
33. Potrafi na bazie frameworka tkinter wskazać podstawowe zastosowanie frameworków.
34. Rozumie zasady planowania i programowania aplikacji na podstawie frameworka.
35. Konwertuje aplikacje zaprogramowane na podstawie danego frameworka do określonego systemu operacyjnego.
36. Tworzenie dokumentacji: rozróżnia różnice między opisywaniem aplikacji w kodach (komentowaniem) a dokumentami opisującymi stworzony kod i instrukcją użytkownika.
37. Rozumie zasady wykonywania testów funkcjonalnych.
38. Rozumie proces testowania i obsługi błędów w języku programowania Python.

I. Sposoby sprawdzania i oceniania osiągnięć edukacyjnych uczniów

Ocenie podlega zarówno wiedza teoretyczna, jak i nabyte w trakcie nauki umiejętności. Ponieważ często spotykamy się z istnieniem kilku rozwiązań tego samego zadania czy problemu, premiowani wysokimi ocenami będą uczniowie proponujący rozwiązania nie omawiane na lekcji, do których doszli w wyniku własnej pomysłowości i dociekliwości albo też dzięki pogłębionemu zaznajomieniu się z omawianym tematem.

Oceniane są:

1. Ćwiczenia, zadania wykonywane na lekcji.
Ocenie podlega:

• wykonanie wszystkich poleceń zgodnie z treścią,
• stopień samodzielności wykonywania zadania,
• pilność,
• końcowy efekt pracy (jakość pracy),
• umiejętność pracy w zespole.

2. Odpowiedzi ustne.
Oceniany jest sposób rozumienia oraz stosowania podstawowych terminów technicznych: najczęściej odpowiedź ustna związana jest z wypowiedziami uczniów w trakcie dyskusji i pracy z dokumentacją techniczną.

3. Aktywność podczas pracy na lekcji.
Ocenie podlega:
• aktywność ucznia w czasie zajęć,
• stopień zaangażowania podczas wykonywania zajęć,
• zainteresowanie tematem lekcji,
• przygotowanie dodatkowych materiałów do lekcji.

4. Kartkówki, sprawdziany pisemne lub praktyczne.
5. Zadania domowe, zadania dodatkowe, prace projektowe.

I. WYMAGANIA NA POSZCZEGÓLNE OCENY:

Ocena niedostateczna

Uczeń:

- nie opanował wiedzy i umiejętności z zakresu materiału programowego,
- nie zna terminologii,
- nie posiada podstawowej znajomości pojęć z dziedziny programowania desktopowego,
- nie potrafi projektować prostych aplikacji desktopowych ,
- nie radzi sobie z programowaniem obiektowym
- nie potrafi zastosować podstawowych algorytmów
- nie potrafi zapisać kodu z e schematu blokowego
- nie potrafi napisać komentarza w aplikacji,
- nie radzi sobie z modyfikowaniem programów,
- nie potrafi zaplanować pracy programistów w zespole,
- nie rozróżnia środowisk programistycznych,
- rozwiązuje testy zawodowe i próbne egzaminy na poziomie poniżej 50%.

Ocena dopuszczająca

Uczeń:

- nie wykazuje zainteresowania przedmiotem,
- posiada minimalny wymagany zasób wiedzy teoretycznej,
- z pomocą nauczyciela, często niezbyt chętnie, posługuje się narzędziami i programami wykorzystując tylko najbardziej podstawowe, wybrane opcje i efekty,

- z dużą pomocą nauczyciela projektuje proste aplikacje desktopowe,
- z dużą pomocą nauczyciela stosuje programowanie obiektowe,
- w niewielkim stopniu potrafi zapisać algorytm jako pseudokod, lista kroków, schemat blokowy,
- z pomocą nauczyciela projektuje algorytm zapisuje go w liście kroków i w schemacie blokowym na koniec z pomocą nauczyciela implementuje,
- w minimalnym stopniu jest w stanie napisać pomocniczy komentarz w implementacji,
- z dużą pomocą nauczyciela modyfikuje program,
- posiada minimalną wiedzę z zakresu planowania ,
- ćwiczenia, prace i projekty wykonuje niestarannie, z dużymi brakami w stosunku do zaprezentowanego przez nauczyciela wzoru lub przykładu, z wykorzystaniem najprostszych opcji i narzędzi,
- rozwiązuje testy zawodowe i próbne egzaminy na poziomie 50-60%.

Ocena dostateczna

Uczeń:

- nie wykazuje zbytniego zainteresowania przedmiotem, niemniej zadane ćwiczenia i prace stara się, mimo trudności, wykonać jak najlepiej,
- w posiadanej wiedzy teoretycznej prezentuje duże braki, niemniej większość materiału ma opanowaną,
- z niewielką pomocą nauczyciela projektuje aplikacje desktopowe,
- z niewielką pomocą nauczyciela stosuje programowanie obiektowe,
- rozumie algorytmy i potrafi, je zapisać przy pomocy listy kroków pseudokodu i schematu blokowego posiłkując się wskazówkami,
- z pomocą nauczyciela projektuje aplikacje i je implementuje posiłkując się notatkami,
- jest w stanie napisać komentarz opisujący algorytm użyty w aplikacji,
- z niedużą pomocą nauczyciela modyfikuje program,
- posiada dostateczną wiedzę z zakresu projektowania aplikacji,
- wykonuje ćwiczenia, prace i projekty z niedbałością, prostotą, brakiem zastosowania wielu opcji i efektów,
- rozwiązuje testy zawodowe i próbne egzaminy na poziomie 60-70%.

Ocena dobra

Uczeń:

- zna w dużym zakresie wymagane pojęcia i terminologię techniczną,
- posiada niewielkie braki w wiedzy teoretycznej przedmiotu,
- projektuje podstawowe aplikacje desktopowe,
- z niewielkimi potknięciami stosuje programowanie obiektowe,
- rozumie algorytmy i potrafi, je zapisać przy pomocy listy kroków pseudokodu i schematu blokowego
- projektuje aplikacje i je implementuje używając samodzielnie programowania obiektowego, niewielka pomoc nauczyciela
- tworzy samodzielnie funkcje, obiekty, potrafi użyć wskaźników. Niewielka pomoc nauczyciela
- z niedużymi błędami jest w stanie napisać komentarz w aplikacji,
- posiada wiedzę w zakresie projektowania aplikacji,
- wykonuje ćwiczenia, prace i projekty z niewielkimi brakami w stosunku do przedstawionego przez nauczyciela wzoru czy przykładu,

- rozwiązuje testy zawodowe i próbne egzaminy na poziomie 70-80%.

Ocena bardzo dobra

Uczeń:

- zna wymagane pojęcia i terminologię techniczną,
- posiada wymaganą na tym etapie nauczania przedmiotu wiedzę teoretyczną,
- posługuje się oprogramowaniem narzędziami i dokumentacją techniczną,
- samodzielnie rozwiązuje prostsze problemy techniczne,
- projektuje aplikacje desktopowe,
- stosuje programowanie obiektowe,
- rozumie algorytmy i potrafi, je zapisać przy pomocy listy kroków pseudokodu i schematu blokowego, potrafi samodzielnie dokonywać modyfikacji,
- projektuje aplikacje i je implementuje używając samodzielnie programowania obiektowego,
- tworzy samodzielnie funkcje, obiekty, potrafi użyć wskaźników, samodzielnie używa zmiennej plikowej,
- Pisze pomocne komentarze w aplikacji,
- posiada szeroką wiedzę z zakresu projektowania aplikacji,
- wykonuje ćwiczenia, prace i projekty z dużą starannością i dokładnością w odtworzeniu zaprezentowanego przez nauczyciela wzoru, przykładu,
- rozwiązuje testy zawodowe i próbne egzaminy na poziomie 80-90%.

Ocena celująca

Uczeń:

- zna wymagane pojęcia i terminologię,
- posiada wymaganą na tym etapie nauczania przedmiotu wiedzę teoretyczną,
- perfekcyjnie i z dużą swobodą posługuje się narzędziami i dokumentacją techniczną,
- samodzielnie rozwiązuje przedstawione na zajęciach problemy techniczne,
- wykonuje ćwiczenia, prace i projekty z dużym stopniem samodzielności i własnej inwencji, złożonością oraz pomysłowością, oryginalnością,
- do swoich prac pozyskuje materiał z bardzo różnych źródeł wiedzy,
- wyróżnia się starannością i solidnością podczas wykonywania powierzonych zadań oraz aktywnością na lekcjach,
- wykazuje ponadprzeciętne zainteresowanie przedmiotem, mogące objawiać się poszerzoną wiedzą i umiejętnościami zdobywanymi we własnym zakresie,
- zdobywa co najmniej wyróżnienia w międzyszkolnych i wyższych konkursach technicznych,
- rozwiązuje testy zawodowe i próbne egzaminy na poziomie powyżej 90%.
- Jest koleżeński pomocny i potrafi zarządzić zespołem i poprowadzić pracami programistycznymi

Znak ‘+’ dodawany jest do oceny podstawowej w ocenianiu bieżącym (oprócz oceny celującej) gdy uczeń nie spełnił wymaganych kryteriów na daną ocenę, a poziom jego wiedzy i umiejętności przekracza według nauczyciela wymagania na ocenę niższą – do 50% wymagań.

Znak ‘-‘ dodawany jest do oceny podstawowej w ocenianiu bieżącym (oprócz oceny niedostatecznej) gdy uczeń według nauczyciela nie spełnił wymaganych kryteriów na daną ocenę- brak 25% wymaganych kryteriów.