



Fundusze
Europejskie
Wiedza Edukacja Rozwój



Rzeczpospolita
Polska

Unia Europejska
Europejski Fundusz Społeczny



PROGRAM NAUCZANIA KURSU UMIEJĘTNOŚCI ZAWODOWYCH

INF.04.4 Programowanie obiektowe

w zakresie kwalifikacji

INF.04 Projektowanie, programowanie i testowanie aplikacji

wyodrębnionej w zawodzie

technik programista 351406

Branża teleinformatyczna (INF)

Warszawa 2021

Autorzy:

mgr inż. Piotr Golonko

mgr Robert Fleischer

Recenzenci:

Recenzent 1 – Recenzja dydaktyczna (nauczyciel uczący w zawodzie, w którym wyodrębniono daną kwalifikację) **mgr inż. Marek Józwiak**

Recenzent 2 – Recenzja merytoryczna (przedstawiciel pracodawców właściwy dla danego zawodu) **mgr Adam Mazgajczyk**

Ekspert:

mgr inż. Grażyna Mrozińska-Hotłoś

Program opracowany we współpracy podmiotów z otoczenia społeczno-gospodarczego wskazanego we wniosku o powierzenie grantu na opracowanie modelowego kwalifikacyjnego kursu zawodowego (KKZ):

DGA S.A. (Partner Wiodący) z Gminą Miastem Toruń (Partner) reprezentowaną przez Toruński Ośrodek Doradztwa Metodycznego i Doskonalenia Osoba prowadząca kursy z Torunia przy współpracy z Edukacja i Kształcenie Zawodowe. EKZ. podmiotami otoczenia społeczno-gospodarczego szkół lub placówek systemu oświaty prowadzących kształcenie zawodowe.

Program Operacyjny Wiedza Edukacja Rozwój

Oś priorytetowa II

Efektywne polityki publiczne dla rynku pracy, gospodarki i edukacji

Działanie 2.14. Rozwój narzędzi dla uczenia się przez całe życie

Konkurs nr POWR.02.14.00-IP.02-00-003/19

Opracowanie modelowych programów kwalifikacyjnych kursów zawodowych (kkz)

Spis treści

PROGRAM NAUCZANIA KURSU UMIEJĘTNOŚCI ZAWODOWYCH INF.04.4 Programowanie obiektowe	4
1. Wprowadzenie	4
2. Plan zajęć kursu umiejętności zawodowych	10
2.1. Pogrupowanie efektów kształcenia	10
2.2. Określenie liczby godzin na kształcenie zawodowe	16
2.3. Plan kursu umiejętności zawodowych	18
3. Cele kształcenia kursu umiejętności zawodowych	18
4. Programy poszczególnych zajęć	18
4.1. Program nauczania dla przedmiotu: Techniki programowania obiektowego	18
4.1.1 Cele ogólne przedmiotu to:	18
4.1.2 Cele szczegółowe przedmiotu (słuchacz/uczestnik potrafi):	19
4.1.3 Materiał nauczania z uwzględnieniem opisu efektów kształcenia	19
4.1.4 Procedury osiągania celów kształcenia	23
4.1.5 Proponowane metody sprawdzania osiągnięć edukacyjnych słuchacza/uczestnika	24
5. Ewaluacja programu kursu dodatkowych umiejętności zawodowych	24
6. Wykaz literatury oraz niezbędnych środków i materiałów dydaktycznych	26
6.1. Wykaz literatury	26
6.2. Wykaz niezbędnych środków i materiałów dydaktycznych	27
7. Sposób i forma zaliczenia kursu	28
8. Sprawdzenie kompletności i poprawności opracowanego programu zajęć	29

PROGRAM NAUCZANIA KURSU UMIEJĘTNOŚCI ZAWODOWYCH INF.04.4 Programowanie obiektowe

1. Wprowadzenie

Charakterystyka kursu umiejętności zawodowych

Kurs może być prowadzony przez:

- publiczne i niepubliczne szkoły prowadzące kształcenie zawodowe, z wyjątkiem szkół artystycznych – w zakresie zawodów, w których kształcą, oraz w zakresie innych zawodów przypisanych do branż, do których należą zawody, w których kształci szkoła,
- publiczne i niepubliczne placówki kształcenia ustawicznego i centra kształcenia zawodowego,
- instytucje rynku pracy, o których mowa w art. 6 ustawy z dnia 20 kwietnia 2004 r. o promocji zatrudnienia i instytucjach rynku pracy, prowadzące działalność edukacyjno-szkoleniową,
- podmioty prowadzące działalność oświatową, o której mowa w art. 170 ust. 2, posiadające akredytację, o której mowa w art. 118. ustawy z dnia 14 grudnia 2016 r. Prawo oświatowe (Dz. U. z 2019 r. poz. 1148, z późn. zm.).

Kurs może być realizowany w formie:

- stacjonarnej – w czasie około 6 miesięcy (1 semestr), łącznie 150 godzin – zajęcia odbywają się 3 lub 4 dni w tygodniu po min. 8 godzin dziennie,
- zaocznej – w czasie około 6 miesięcy (65% z 150 godzin = 98 godzin) – zajęcia odbywają się co 2 tygodnie przez 2 dni po 10 godzin dziennie, a w uzasadnionych przypadkach – co tydzień przez 2 dni po 10 godzin dziennie.
- efekty kształcenia wskazane do realizacji w kształceniu teoretycznym mogą być (po spełnieniu wymagań określonych w aktualnych przepisach oświatowych) realizowane w formie kształcenia na odległość, przy czym zaliczenie tych zajęć nie może odbywać się w formie zdalnej. Przedmioty o kształceniu praktycznym ze względu na efekty uczenia wymagające fizycznej interakcji muszą być prowadzone w formie stacjonarnej.

Informacje dodatkowe:

- Kurs jest prowadzony na poziomie IV Polskiej Ramy Kwalifikacji
- Kurs nie jest związany z szczególnymi uwarunkowaniami związanymi z kształceniem w kwalifikacji.
- Ośrodek prowadzący kurs ma obowiązek zgłoszenia okręgowej komisji egzaminacyjnej informacji o jego rozpoczęciu (w przeciągu 14 dni od rozpoczęcia realizacji KKZ).

Kurs jest przeznaczony dla osób chcących:

- zdobyć nowy zawód
- uzupełnić swoje wykształcenie
- udoskonalić swoje umiejętności
- wspomóc rozwój swojej kariery zawodowej
- zwiększyć szanse na znalezienie pracy
- dokonać zmiany pracy
- uzyskać awans zawodowy

- utrzymać zatrudnienie.

Wymagania wstępne:

- pozytywne przejście badań lekarskich (brak przeciwwskazań lekarskich do odbycia kursu)

Sposób i forma zaliczenia kursu:

- Kwalifikacyjny kurs zawodowy kończy się zaliczeniem w formie walidacji osiągnięć uczestnika kursu, polegającej na ocenie wykonywanych w trakcie nauki projektów i ćwiczeń oraz na podstawie uzyskanych w trakcie kursu ocen z poszczególnych przedmiotów.
- Do oceny osiągnięć edukacyjnych słuchaczy proponuje się stosowanie testów wielokrotnego wyboru, zadań z luką, ocenę aktywności słuchacza podczas wykonywania zadań w grupie, ocenę jakości wykonania zadań przez słuchacza. Proponuje się, aby osiągnięcia słuchaczy oceniać w zakresie zaplanowanych, uszczegółowionych celów kształcenia na podstawie:
 - obserwacji wykonanych ćwiczeń,
 - testu pisemnego.
- Umiejętności praktyczne proponuje się sprawdzać na podstawie obserwacji czynności wykonywanych przez słuchacza w trakcie realizacji ćwiczeń. Podczas obserwacji należy zwrócić uwagę na:
 - wyszukiwanie i przetwarzanie rzetelnych informacji pozyskanych z różnych źródeł,
 - poprawność merytoryczną wykonanych ćwiczeń praktycznych,
 - umiejętność pracy w zespole.
- Ważne kryteria oceny efektów kształcenia to: zaplanowanie wykonania zadania, dobór elementów oraz sporządzona dokumentacja techniczna. Możliwe są również inne sposoby i formy zaliczenia, takie jak: testy praktyczne, wykonanie projektów, próby pracy, aktywność uczącego się na zajęciach, prezentacje na forum grupy z przeprowadzonych prac.
- Osoba, która uzyskała zaliczenie, otrzymuje zaświadczenie o ukończeniu kursu dodatkowych umiejętności zawodowych.
- Kurs musi się zakończyć co najmniej 6 tygodni przed planowanym egzaminem z kwalifikacji zawodowej.

Założenia programowe

Głównym celem kształcenia w technik programista jest przygotowanie szeroko wykwalifikowanej kadry specjalistów. Przygotowanych do:

- profesjonalnego i rzetelnego wykonywania czynności zawodowych,
- pracy w ciągle zmieniającej się rzeczywistości zawodowej,
- samodzielnego podnoszenia swoich kwalifikacji,
- podejmowania własnej działalności gospodarczej zgodnej z zawodem,
- pracy w zespole,
- szybkiej aktualizacji wiedzy z niezwykle dynamicznej dziedziny, jaką jest branża teleinformatyczna,
- kontynuowania edukacji w szkołach wyższych na kierunkach: elektronika, teleinformatyka, telekomunikacja, informatyka lub zbliżonych.

Struktura programu

Program nauczania jest o strukturze przedmiotowej i spiralnej w układzie treści, z układem materiału nauczania zaczynającym się od zagadnień najprostszych po trudniejsze. Taki układ umożliwia powrót do treści zrealizowanych na początku edukacji, aby je powtórzyć i poszerzyć w kolejnych latach nauki. Utrwala to zarówno wiedzę jak i nabywane umiejętności celem przygotowania do realizacji zadań zawodowych. Dodatkowo taki układ i cykl nauczania w znaczącym stopniu niweluje braki edukacyjne, oraz pozwala na analizę materiału nauczania przez słuchaczy na różnych poziomach umiejętności.

Cele kierunkowe

Absolwent kursu powinien być przygotowany do wykonywania następujących zadań zawodowych w zakresie:

- tworzenia stron i aplikacji internetowych,
- tworzenia i zarządzania bazami danych,
- tworzenia aplikacji desktopowych,
- tworzenia aplikacji mobilnych,
- testowania i dokumentowania aplikacji.

Rozkład treści nauczania uwzględnia wzajemną korelację pomiędzy przedmiotami, a kolejność zdobywania wiedzy i umiejętności pozwala na nabycie wiedzy teoretycznej, by w krótkim czasie wykorzystać ją praktycznie. Zajęcia są realizowane na przedmiotach kształcenia teoretycznego oraz praktycznego. Dzięki czemu po zakończeniu kursu absolwent potrafi:

- projektować i tworzyć aplikacje desktopowe, mobilne i webowe,
- korzystać z bibliotek i frameworków w czasie tworzenia i modyfikowania aplikacji,
- zarządzać projektami informatycznymi,
- wdrażać aplikacje,
- wdrażać poprawki aplikacji,
- wdrażać nowe wersje aplikacji,
- wdrażać nowe funkcjonalności,
- współpracować z zespołem programistów,
- współpracować z klientem,
- planować testy aplikacji,
- testować aplikacje,
- tworzyć dokumentację z przeprowadzonych testów,
- tworzyć dokumentację zgłoszonych błędów,
- tworzyć dokumentację zapotrzebowania na funkcjonalności,
- tworzyć dokumentację programu,
- tworzyć dokumentację wdrożeniową,
- tworzyć dokumentację (instrukcje) użytkownika aplikacji.

Opis branży

Branża teleinformatyczna będąc jedną z wielu branż przyszłości składa się z szeregu zawodów ukierunkowanych na świadczenie usług związanych z budową i eksploatacją systemów teleinformatycznych. Branża ta obejmuje szeroki zakres umiejętności i zawodów. Od umiejętności monterskich związanych z mechanicznym i elektrycznym montażem kabli i urządzeń poprzez ich konfigurację i eksploatację aż po tworzenie programów, zarządzania infrastrukturą informatyczną czy też bazami danych. W części tych zawodów bardzo ważne są kompetencje społeczne zwłaszcza w pracy z osobami niepełnosprawnymi. Branża cechuje się dużą dynamiką rozwoju i ciągłym zapotrzebowaniem na pracowników spowodowanym ciągłym i dynamicznym rozwojem usług i technologii przesyła i przetwarzania informacji.

Odniesienie do rynku pracy

Zawód technik programista 351406, który powstał w oparciu o rekomendacje pracodawców zgłaszających zapotrzebowanie na wykwalifikowanych kandydatów do pracy. Jest nowoczesnym i wymagającym zawodem przyszłości, stawiającym ciągle nowe wyzwania i dającym możliwości samorealizacji i dużej satysfakcji z wykonywanej pracy. Branża teleinformatyczna dynamicznie się rozwija dając możliwość samorozwoju i zapewniając liczne miejsca pracy. W związku z tym istnieje zapotrzebowanie na osoby wykwalifikowane w tym zawodzie. Pracodawcy oczekują absolwenta wyposażonego w wiele kluczowych umiejętności i potrafiącego szybko reagować na zmieniającą się rzeczywistość oraz pogłębiać swoją wiedzę i umiejętności w zakresie nowych rozwiązań konstrukcyjnych i technologii.

Głównym celem kształcenia w zawodzie technik programista jest przygotowanie wykwalifikowanej kadry specjalistów do pracy w dynamicznie zmieniającym się sektorze gospodarki jakim jest branża teleinformatyczna.

Osoba posiadająca wykształcenie w tym zawodzie może pracować:

- przedsiębiorstwa zajmujące się tworzeniem programów desktopowych i aplikacji internetowych;
- przedsiębiorstwa zajmujące się tworzeniem aplikacji mobilnych;
- przedsiębiorstwa zajmujące się projektowaniem UI;
- przedsiębiorstwa zajmujące się modelowaniem, projektowaniem i drukiem 3D.

Poza swoją branżą technik programista 351406 może pracować na stanowiskach związanych z montażem elektrycznym, elektronicznym i mechanicznym sieci i urządzeń teletransmisyjnych związanych z szerokopasmową transmisją danych oraz wykonywaniem instalacji urządzeń teletransmisyjnych i ich uruchamianiem. Do podjęcia pracy w tym zawodzie niezbędna jest wysoka sprawność manualna i dobra koordynacja wzrokowo - ruchowa.

Pracodawcy powinni aktywnie uczestniczyć w cyklu kształcenia poprzez:

- konferencje branżowe,
- seminaria,
- lekcje problemowe,
- lekcje pokazowe,
- praktyki zawodowe,

organizowane wspólnie z szkołą.

Powiązanie z zawodami

Branża teleinformatyczna pokrywa znaczny zakres usług, technik programista 351406 świadczy usługi z pogranicza telekomunikacji i informatyki, które są ukierunkowane na montaż, uruchamianie i utrzymanie sieci teleinformatycznych oraz instalowanie i administrowanie sieciowymi systemami komputerami. Umiejętności praktyczne są wzbogacone wiedzą ogólną z zakresu techniki analogowej i cyfrowej oraz wiedzą szczegółową z zakresu systemów komputerowych, sieciowych systemów operacyjnych, lokalnych sieci komputerowych, urządzeń sieciowych, systemów transmisji danych i systemów komutacyjnych.

Zawód technik programista wiąże się z szeregiem zawodów o zbliżonej tematyce, zakresie wiedzy i umiejętności. Umożliwia to zarówno pogłębianie wiedzy jak i łatwe przebranżowienie się na jeden z zawodów pokrewnych:

- technik teleinformatyk,
- technik informatyk,
- technik szerokopasmowej komunikacji elektronicznej,
- technik telekomunikacji,
- monter sieci i urządzeń telekomunikacyjnych,
- technik tyfloinformatyk.

Dodatkowo technik programista może też się sprawdzić w zawodach z branży związanej z elektroniką takich jak:

- elektronik,
- technik elektronik.

Informacje o pozostałych Kursach Umiejętności Zawodowych

Istnieje możliwość uzupełnienia wiedzy i umiejętności zawodowych poprzez udział w dodatkowych Kursach Umiejętności Za dedykowanych dla zawodu technik programista 351406 (opisanych w odrębnych plikach):

- Bezpieczeństwo sieci komputerowych;
- Bezpieczeństwo systemów komputerowych;
- Budowa i konfiguracja sieci komputerowych;
- Eksploatacja baz danych;
- Grafika 3D i wydruk 3D;
- Programowanie mikrokontrolerów i prostych układów scalonych;
- Programowanie w języku Python;
- Serwis urządzeń techniki komputerowej;
- Tworzenie i testowanie aplikacji.

Po ukończeniu KUZ absolwent może realizować program kwalifikacyjnego kursu zawodowego INF.04 Projektowanie, programowanie i testowanie aplikacji opartego o podstawę programową kształcenia branżowego w zawodzie technik programista 351406, w której to wyodrębniono dla kwalifikacji INF.04 Projektowanie, programowanie i testowanie aplikacji następujące jednostki efektów kształcenia:

- INF.04.1. Bezpieczeństwo i higiena pracy,

- INF.04.2. Podstawy informatyki,
- INF.04.3. Projektowanie oprogramowania,
- INF.04.4. Programowanie obiektowe,
- INF.04.5. Programowanie aplikacji desktopowych,
- INF.04.6. Programowanie aplikacji mobilnych,
- INF.04.7. Programowanie aplikacji zaawansowanych webowych,
- INF.04.8. Testowanie i dokumentowanie aplikacji,
- INF.04.9. Język obcy zawodowy.

oraz efekty kształcenia realizowane na wszystkich obowiązkowych zajęć edukacyjnych z zakresu kształcenia zawodowego związane z nabywaniem kompetencji personalnych i społecznych i organizacji pracy małych zespołów, zgrupowane w jednostkach efektów kształcenia:

- INF.04.10. Kompetencje personalne i społeczne,
- INF.04.11. Organizacja pracy małych zespołów

Po ukończeniu KUZ absolwent może realizować programy kursów umiejętności zawodowych w obrębie kwalifikacji INF.04 Projektowanie, programowanie i testowanie aplikacji opartego o podstawę programową kształcenia branżowego w zawodzie technik programista 351406, w której to wyodrębniono następujące jednostki efektów kształcenia (opisanych w odrębnych plikach):

- INF.04.1. Bezpieczeństwo i higiena pracy,
- INF.04.2. Podstawy informatyki,
- INF.04.3. Projektowanie oprogramowania,
- INF.04.5. Programowanie aplikacji desktopowych,
- INF.04.6. Programowanie aplikacji mobilnych,
- INF.04.7. Programowanie aplikacji zaawansowanych webowych,
- INF.04.8. Testowanie i dokumentowanie aplikacji,
- INF.04.9. Język obcy zawodowy.

2. Plan zajęć kursu umiejętności zawodowych

2.1. Pogrupowanie efektów kształcenia

Tabela 1. Przyporządkowanie efektów kształcenia wraz z kryteriami weryfikacji do poszczególnych przedmiotów.

Efekty kształcenia Stopniowane efektów kształcenia efekt kluczowy ek,, efekt ważny ew, efekt pomocniczy ep	Liczba godzin na efekt kształcenia	Kryteria weryfikacji	Techniki programowania
A	B	C	
wykorzystuje środowisko programistyczne dla obiektowych aplikacji konsolowych (ew)	30	rozróżnia kompilatory i interpretery	x
		charakteryzuje zadania kompilatora, interpretera, debuggera	x
		analizuje błędy w kodzie za pomocą debuggera	x
		charakteryzuje etapy kompilacji i interpretacji kodu	x
		charakteryzuje pojęcie biblioteki	x
		kompiluje i uruchamia programy	x
przestrzega zasad programowania (ew)	15	dzieli program na funkcje (metody)	x
		stosuje rekurencję	x
		implementuje algorytmy w programie	x
korzysta z typów danych (ek)	16	stosuje proste i złożone typy danych	x
		deklaruje własne typy danych	x
		deklaruje zmienne różnych typów danych	x
		wykonuje operacje na zmiennych: wejścia i wyjścia, arytmetyczne, logiczne	x
		stosuje typy złożone i operacje na nich	x
stosuje wyrażenia, instrukcje i biblioteki (ek)	15	stosuje operatory arytmetyczne, przypisania, porównania, logiczne, operatory do obsługi łańcuchów, bitowe	x
		wykorzystuje priorytety operatorów do właściwego budowania wyrażeń	x
		stosuje instrukcję warunkową i wyboru	x
		stosuje instrukcje pętli	x



Efekty kształcenia Stopniowane efektów kształcenia efekt kluczowy ek,, efekt ważny ew, efekt pomocniczy ep	Liczba godzin na efekt kształcenia	Kryteria weryfikacji	Techniki programowania
A	B	C	
		korzysta z wybranych bibliotek języka C++, C#, Python lub innego języka programowania: biblioteka standardowa, biblioteka z funkcjami matematycznymi, biblioteka z podstawowymi algorytmami	x
stosuje zasady programowania obiektowego (ek)	20	stosuje obiektowe podejście do rozwiązywania problemów	x
		charakteryzuje pojęcia klasa, obiekt, metoda, pole, dziedziczenie, hermetyzacja, polimorfizm	x
		dzieli zagadnienie na klasy	x
		powołuje obiekty	x
		planuje aplikację z zastosowaniem hermetyzacji, dziedziczenia i polimorfizmu	
definiuje klasy (ek)	20	definiuje pola klasy	x
		określa zakres widoczności pól klasy i definiuje kwalifikatory dostępu	x
		definiuje metody klasy	x
		definiuje konstruktory, w tym konstruktor kopiujący, i destruktor klasy	x
		definiuje instrukcje inicjujące konstruktora	x
		określa zakres widoczności metod klasy i definiuje kwalifikatory dostępu	x
		implementuje funkcjonalność klasy	x
		deklaruje obiekty i odwołuje się obiektem do składowych klasy	x
		definiuje składniki statyczne klasy	x
		rozdziela klasy dziedziczone i zaprzyjaźnione	x
		tworzy funkcje zaprzyjaźnione z klasą	x
		stosuje składnik statyczny klasy i metody do ich obsługi	x
definiuje klasy pochodne (ew)	14	buduje hierarchię dziedziczenia klas w programie	x
		wydziała metody i pola do odpowiednich klas w hierarchii dziedziczenia	x
		definiuje klasy bazowe i pochodne	x



Efekty kształcenia Stopniowane efektów kształcenia efekt kluczowy ek,, efekt ważny ew, efekt pomocniczy ep	Liczba godzin na efekt kształcenia	Kryteria weryfikacji	Techniki programowania
A	B	C	
		stosuje metody wirtualne, definiuje klasy abstrakcyjne	x
programuje szablony (wzorce) klas (ew)	10	definiuje szablony klas dla obsługi prostych typów liczbowych	x
programuje obsługę wyjątków (ew)	10	stosuje szkielet obsługi wyjątków z instrukcjami try i catch	x
		stosuje instrukcję throw	x
		opracowuje listę możliwych błędów wykonania aplikacji	x
		definiuje obsługę dla błędów wykonania aplikacji	x
Razem liczba godzin w jednostce efektów kształcenia	150		

Nauczyciele wszystkich obowiązkowych zajęć edukacyjnych powinni stwarzać warunki do nabywania kompetencji personalnych i społecznych w ramach godzin przeznaczonych na kształcenie zawodowe.

Tabela 2. Grupowanie efektów kształcenia i nadawanie nazw tym zajęciom.

Nazwa jednostki efektów kształcenia	Efekt kształcenia wraz z kodowaniem (ek, ew, ep)	Kryteria weryfikacji kryteria podstawowe kryteria ponadpodstawowe	Grupowanie efektów kształcenia w przedmioty	Liczba godzin na poszczególne efekty kształcenia	Okres realizacji w cyklu nauczania
A	B	C	D	E	F
INF.04.4. Programowanie obiektowe (150h)	wykorzystuje środowisko programistyczne dla obiektowych aplikacji konsolowych (ew)	rozdziela kompilatory i interpretery	Techniki programowania obiektowego (150h)	30	W czasie kursu
		charakteryzuje zadania kompilatora, interpretera, debuggera			
		analizuje błędy w kodzie za pomocą debuggera			
		charakteryzuje etapy kompilacji i interpretacji kodu			

Nazwa jednostki efektów kształcenia	Efekt kształcenia wraz z kodowaniem (ek, ew, ep)	Kryteria weryfikacji kryteria podstawowe kryteria ponadpodstawowe	Grupowanie efektów kształcenia w przedmioty	Liczba godzin na poszczególne efekty kształcenia	Okres realizacji w cyklu nauczania
A	B	C	D	E	F
		charakteryzuje pojęcie biblioteki			
		kompiluje i uruchamia programy			
	przestrzega zasad programowania (ew)	dzieli program na funkcje (metody)		15	
		stosuje rekurencję			
		implementuje algorytmy w programie			
	korzysta z typów danych (ek)	stosuje proste i złożone typy danych		16	
		deklaruje własne typy danych			
		deklaruje zmienne różnych typów danych			
		wykonuje operacje na zmiennych: wejścia i wyjścia, arytmetyczne, logiczne			
		stosuje typy złożone i operacje na nich			
	stosuje wyrażenia, instrukcje i biblioteki (ek)	stosuje operatory arytmetyczne, przypisania, porównania, logiczne, operatory do obsługi łańcuchów, bitowe		15	
		wykorzystuje priorytety operatorów do właściwego budowania wyrażeń			
		stosuje instrukcję warunkową i wyboru			
		stosuje instrukcje pętli			

Nazwa jednostki efektów kształcenia	Efekt kształcenia wraz z kodowaniem (ek, ew, ep)	Kryteria weryfikacji kryteria podstawowe kryteria ponadpodstawowe	Grupowanie efektów kształcenia w przedmioty	Liczba godzin na poszczególne efekty kształcenia	Okres realizacji w cyklu nauczania
A	B	C	D	E	F
		korzysta z wybranych bibliotek języka C++, C#, Python lub innego języka programowania: biblioteka standardowa, biblioteka z funkcjami matematycznymi, biblioteka z podstawowymi algorytmami			
	stosuje zasady programowania obiektowego (ek)	stosuje obiektowe podejście do rozwiązywania problemów		20	
		charakteryzuje pojęcia klasa, obiekt, metoda, pole, dziedziczenie, hermetyzacja, polimorfizm			
		dzieli zagadnienie na klasy			
		powołuje obiekty			
	definiuje klasy (ek)	definiuje pola klasy		20	
		określa zakres widoczności pól klasy i definiuje kwalifikatory dostępu			
		definiuje metody klasy			
		definiuje konstruktory, w tym konstruktor kopiujący, i destruktor klasy			
		definiuje instrukcje inicjujące konstruktora			
		określa zakres widoczności metod klasy i definiuje kwalifikatory dostępu			
		implementuje funkcjonalność klasy			

Nazwa jednostki efektów kształcenia	Efekt kształcenia wraz z kodowaniem (ek, ew, ep)	Kryteria weryfikacji kryteria podstawowe kryteria ponadpodstawowe	Grupowanie efektów kształcenia w przedmioty	Liczba godzin na poszczególne efekty kształcenia	Okres realizacji w cyklu nauczania
A	B	C	D	E	F
		deklaruje obiekty i odwołuje się obiektem do składowych klasy			
		definiuje składniki statyczne klasy			
		rozróżnia klasy dziedziczone i zaprzyjaźnione			
		tworzy funkcje zaprzyjaźnione z klasą			
		stosuje składnik statyczny klasy i metody do ich obsługi			
	definiuje klasy pochodne (ew)	buduje hierarchię dziedziczenia klas w programie		14	
		wydziela metody i pola do odpowiednich klas w hierarchii dziedziczenia			
		definiuje klasy bazowe i pochodne			
		stosuje metody wirtualne, definiuje klasy abstrakcyjne			
	programuje szablony (wzorce) klas (ew)	definiuje szablony klas dla obsługi prostych typów liczbowych		10	
	programuje obsługę wyjątków (ew)	stosuje szkielet obsługi wyjątków z instrukcjami try i catch		10	
		stosuje instrukcję throw			
		opracowuje listę możliwych błędów wykonania aplikacji			
		definiuje obsługę dla błędów wykonania aplikacji			



2.2. Określenie liczby godzin na kształcenie zawodowe

Tabela 3. Określenie liczby godzin poszczególnych zajęć z podziałem na zajęcia teoretyczne i praktyczne

Przedmiot	Liczba godzin zajęć teoretycznych	Liczba godzin zajęć praktycznych	Efekty kształcenia	Kryteria weryfikacji
A	B	C	D	E
Techniki programowania obiektowego (150h)		3	wykorzystuje środowisko programistyczne dla obiektowych aplikacji konsolowych (ew)	rozdziela kompilatory i interpretery
				charakteryzuje zadania kompilatora, interpretera, debuggera
				analizuje błędy w kodzie za pomocą debuggera
				charakteryzuje etapy kompilacji i interpretacji kodu
				charakteryzuje pojęcie biblioteki
				kompiluje i uruchamia programy
		15	przestrzega zasad programowania (ew)	dzieli program na funkcje (metody)
				stosuje rekurencję
				implementuje algorytmy w programie
		16	korzysta z typów danych (ek)	stosuje proste i złożone typy danych
				deklaruje własne typy danych
				deklaruje zmienne różnych typów danych
				wykonuje operacje na zmiennych: wejścia i wyjścia, arytmetyczne, logiczne
				stosuje typy złożone i operacje na nich
		15	stosuje wyrażenia, instrukcje i biblioteki (ek)	stosuje operatory arytmetyczne, przypisania, porównania, logiczne, operatory do obsługi łańcuchów, bitowe
				wykorzystuje priorytety operatorów do właściwego budowania wyrażeń
				stosuje instrukcję warunkową i wyboru
				stosuje instrukcje pętli
				korzysta z wybranych bibliotek języka C++, C#, Python lub innego języka programowania: biblioteka standardowa, biblioteka z funkcjami matematycznymi, biblioteka z podstawowymi algorytmami
		2	stosuje zasady programowania obiektowego (ek)	stosuje obiektowe podejście do rozwiązywania problemów
				charakteryzuje pojęcia klasa, obiekt, metoda, pole, dziedziczenie, hermetyzacja, polimorfizm



Przedmiot	Liczba godzin zajęć teoretycznych	Liczba godzin zajęć praktycznych	Efekty kształcenia	Kryteria weryfikacji
A	B	C	D	E
				dzieli zagadnienie na klasy
				powołuje obiekty
		2	definiuje klasy (ek)	definiuje pola klasy
				określa zakres widoczności pól klasy i definiuje kwalifikatory dostępu
				definiuje metody klasy
				definiuje konstruktory, w tym konstruktor kopiujący, i destruktor klasy
				definiuje instrukcje inicjujące konstruktora
				określa zakres widoczności metod klasy i definiuje kwalifikatory dostępu
				implementuje funkcjonalność klasy
				deklaruje obiekty i odwołuje się obiektem do składowych klasy
				definiuje składniki statyczne klasy
				rozróżnia klasy dziedziczone i zaprzyjaźnione
				tworzy funkcje zaprzyjaźnione z klasą
				stosuje składnik statyczny klasy i metody do ich obsługi
		14	definiuje klasy pochodne (ew)	buduje hierarchię dziedziczenia klas w programie
				wydziela metody i pola do odpowiednich klas w hierarchii dziedziczenia
				definiuje klasy bazowe i pochodne
				stosuje metody wirtualne, definiuje klasy abstrakcyjne
		1	programuje szablony (wzorce) klas (ew)	definiuje szablony klas dla obsługi prostych typów liczbowych
		1	programuje obsługę wyjątków (ew)	stosuje szkielet obsługi wyjątków z instrukcjami try i catch
				stosuje instrukcję throw
				opracowuje listę możliwych błędów wykonania aplikacji
				definiuje obsługę dla błędów wykonania aplikacji
Razem godzin:	0	150		
SUMA:	150			

2.3. Plan kursu umiejętności zawodowych

Kurs może być realizowany w formie:

- stacjonarnej – w czasie około 6 miesięcy (1 semestr), łącznie 150 godzin – zajęcia odbywają się 3 lub 4 dni w tygodniu po min. 8 godzin dziennie,
- zaocznej – w czasie około 6 miesięcy (65% z 150 godzin = 98 godzin) – zajęcia odbywają się co 2 tygodnie przez 2 dni po 10 godzin dziennie, a w uzasadnionych przypadkach – co tydzień przez 2 dni po 10 godzin dziennie.
- efekty kształcenia wskazane do realizacji w kształceniu teoretycznym mogą być (po spełnieniu wymagań określonych w aktualnych przepisach oświatowych) realizowane w formie kształcenia na odległość, przy czym zaliczenie tych zajęć nie może odbywać się w formie zdalnej. Przedmioty o kształceniu praktycznym ze względu na efekty uczenia wymagające fizycznej interakcji muszą być prowadzone w formie stacjonarnej.

Tabela 4. Plan zajęć kursu dodatkowych umiejętności zawodowych

Lp.	Powiązanie z podstawą programową	Przedmioty	Liczba godzin	Uwagi
Przedmioty praktyczne				Ze względu, iż w ramach kursu jest jeden przedmiot należy zwracać szczególną uwagę przy omawianiu tematyki zajęć na podstawie teoretyczną oraz zachęcać do korzystania z dokumentacji technicznej
1	INF.04.4. Programowanie obiektowe	Techniki programowania obiektowego	150	
		Łączna liczba godzin:	150	

3. Cele kształcenia kursu umiejętności zawodowych

Absolwent kursu powinien być przygotowany do wykonywania następujących zadań zawodowych w zakresie:

- programowanie obiektowe,
- stosowanie bibliotek,
- obsługa błędów,
- obsługa dziedziczenia i hierarchii.

4. Programy poszczególnych zajęć

4.1. Program nauczania dla przedmiotu: Techniki programowania obiektowego

4.1.1 Cele ogólne przedmiotu to:

- Programowanie obiektowe.
- Stosowanie bibliotek.
- Obsługiwanie błędów.
- Programowanie dziedziczenia i hierarchii.

- Nabycie kompetencji personalnych i społecznych.
- Nabywanie umiejętności pracy w grupach i zespołach.

4.1.2 Cele szczegółowe przedmiotu (słuchacz/uczestnik potrafi):

- stosować różne typy danych,
- kompilować kod programu,
- debiutować kod programu,
- obsługiwać błędy i wyjątki w programie,
- stosować biblioteki programistyczne,
- tworzyć funkcje,
- tworzyć klasy,
- tworzyć obiekty,
- określać zasięg zmiennych,
- zarządzać hierarchią dziedziczenia,
- implementować algorytmy,
- wyszukiwać i poprawiać błędy.

4.1.3 Materiał nauczania z uwzględnieniem opisu efektów kształcenia

Dział programowy	Tematy jednostek metodycznych	liczba godzin	Opis efektów kształcenia uwzględniający kryteria weryfikacji
Wstęp do programowania	Środowiska programistyczne, programowanie strukturalne	10	<ul style="list-style-type: none"> — rozróżnia kompilatory i interpretery — charakteryzuje zadania kompilatora, interpretera, debuggera — charakteryzuje etapy kompilacji i interpretacji kodu — kompiluje i uruchamia programy — stosuje rekurencję — stosuje proste i złożone typy danych — deklaruje własne typy danych — deklaruje zmienne różnych typów danych — stosuje operatory arytmetyczne, przypisania, porównania, logiczne, operatory do obsługi łańcuchów, bitowe — stosuje instrukcję warunkową i wyboru — stosuje instrukcje pętli — analizuje błędy w kodzie za pomocą debuggera

Dział programowy	Tematy jednostek metodycznych	liczba godzin	Opis efektów kształcenia uwzględniający kryteria weryfikacji
			<ul style="list-style-type: none"> — dzieli program na funkcje (metody) — implementuje algorytmy w programie — wykonuje operacje na zmiennych: wejścia i wyjścia, arytmetyczne, logiczne — stosuje typy złożone i operacje na nich — wykorzystuje priorytety operatorów do właściwego budowania wyrażeń
Wstęp do programowania	Programowanie z wykorzystaniem bibliotek	10	<ul style="list-style-type: none"> — rozróżnia kompilatory i interpretery — charakteryzuje zadania kompilatora, interpretera, debuggera — charakteryzuje etapy kompilacji i interpretacji kodu — charakteryzuje pojęcie biblioteki — kompiluje i uruchamia programy — stosuje operatory arytmetyczne, przypisania, porównania, logiczne, operatory do obsługi łańcuchów, bitowe — stosuje instrukcję warunkową i wyboru — stosuje instrukcje pętli — analizuje błędy w kodzie za pomocą debuggera — wykorzystuje priorytety operatorów do właściwego budowania wyrażeń — korzysta z wybranych bibliotek języka C++, C#, Python lub innego języka programowania: biblioteka standardowa, biblioteka z funkcjami matematycznymi, biblioteka z podstawowymi algorytmami
Wstęp do programowania	Niezawodność oprogramowania	20	<ul style="list-style-type: none"> — kompiluje i uruchamia programy — stosuje instrukcję throw — definiuje obsługę dla błędów wykonania aplikacji — analizuje błędy w kodzie za pomocą debuggera — stosuje szkielet obsługi wyjątków z instrukcjami try i catch — opracowuje listę możliwych błędów wykonania aplikacji
Programowanie obiektowe	Wstęp do programowania obiektowego	30	<ul style="list-style-type: none"> — rozróżnia kompilatory i interpretery — charakteryzuje zadania kompilatora, interpretera, debuggera — charakteryzuje pojęcie biblioteki — kompiluje i uruchamia programy — stosuje rekurencję — stosuje proste i złożone typy danych

Dział programowy	Tematy jednostek metodycznych	liczba godzin	Opis efektów kształcenia uwzględniający kryteria weryfikacji
			<ul style="list-style-type: none"> — deklaruje własne typy danych — deklaruje zmienne różnych typów danych — stosuje operatory arytmetyczne, przypisania, porównania, logiczne, operatory do obsługi łańcuchów, bitowe — stosuje instrukcję warunkową i wyboru — stosuje instrukcje pętli — stosuje obiektowe podejście do rozwiązywania problemów — charakteryzuje pojęcia klasa, obiekt, metoda, pole, dziedziczenie, hermetyzacja, polimorfizm — powołuje obiekty — definiuje pola klasy — określa zakres widoczności pól klasy i definiuje kwalifikatory dostępu — definiuje metody klasy — definiuje konstruktory, w tym konstruktor kopiujący, i destruktor klasy — definiuje instrukcje inicjujące konstruktora — definiuje składniki statyczne klasy — analizuje błędy w kodzie za pomocą debuggera — implementuje algorytmy w programie — wykonuje operacje na zmiennych: wejścia i wyjścia, arytmetyczne, logiczne — stosuje typy złożone i operacje na nich — wykorzystuje priorytety operatorów do właściwego budowania wyrażeń — korzysta z wybranych bibliotek języka C++, C#, Python lub innego języka programowania: biblioteka standardowa, biblioteka z funkcjami matematycznymi, biblioteka z podstawowymi algorytmami — dzieli zagadnienie na klasy — określa zakres widoczności metod klasy i definiuje kwalifikatory dostępu — implementuje funkcjonalność klasy — deklaruje obiekty i odwołuje się obiektem do składowych klasy
Programowanie obiektowe	Programowanie obiektowe	40	<ul style="list-style-type: none"> — charakteryzuje pojęcie biblioteki — kompiluje i uruchamia programy — definiuje pola klasy

Dział programowy	Tematy jednostek metodycznych	liczba godzin	Opis efektów kształcenia uwzględniający kryteria weryfikacji
			<ul style="list-style-type: none"> — określa zakres widoczności pól klasy i definiuje kwalifikatory dostępu — definiuje metody klasy — definiuje konstruktory, w tym konstruktor kopiujący, i destruktor klasy — definiuje instrukcje inicjujące konstruktora — definiuje składniki statyczne klasy — analizuje błędy w kodzie za pomocą debuggera — określa zakres widoczności metod klasy i definiuje kwalifikatory dostępu — implementuje funkcjonalność klasy — deklaruje obiekty i odwołuje się obiektem do składowych klasy
Programowanie obiektowe	Dziedziczenie i hierarchia	40	<ul style="list-style-type: none"> — kompiluje i uruchamia programy — definiuje pola klasy — określa zakres widoczności pól klasy i definiuje kwalifikatory dostępu — definiuje metody klasy — definiuje konstruktory, w tym konstruktor kopiujący, i destruktor klasy — definiuje instrukcje inicjujące konstruktora — definiuje składniki statyczne klasy — rozróżnia klasy dziedziczone i zaprzyjaźnione — stosuje składnik statyczny klasy i metody do ich obsługi — wydziela metody i pola do odpowiednich klas w hierarchii dziedziczenia — definiuje klasy bazowe i pochodne — definiuje szablony klas dla obsługi prostych typów liczbowych — analizuje błędy w kodzie za pomocą debuggera — określa zakres widoczności metod klasy i definiuje kwalifikatory dostępu — implementuje funkcjonalność klasy — deklaruje obiekty i odwołuje się obiektem do składowych klasy — tworzy funkcje zaprzyjaźnione z klasą — buduje hierarchię dziedziczenia klas w programie — stosuje metody wirtualne, definiuje klasy abstrakcyjne

4.1.4 Procedury osiągnięcia celów kształcenia

Propozycje metod nauczania

- metoda projektu (zalecana),
- metoda tekstu przewodniego,
- symulacje,
- gry dydaktyczne,
- pokaz z objaśnieniem,
- pokaz z instruktażem,
- ćwiczenia praktyczne (zalecana),
- praca w grupach (zalecana),
- użycie platform dydaktycznych,
- nauczanie na odległość z wykorzystaniem platform edukacyjnych.

Obudowa dydaktyczna

stanowisko komputerowe dla osoba prowadzącej kurs z:

- dostępem do internetu,
- pakietem programów biurowych
- projektorem multimedialnym lub tablicą multimedialną,
- urządzenie wielofunkcyjne sieciowe (drukarka z skanerem),
- monitorem minimum 21",
- zainstalowanymi środowiskami programistycznymi,

stanowisko komputerowe dla kursanta (1 stanowisko na osobę) z:

- dostępem do internetu,
- pakietem programów biurowych
- monitorem minimum 21",
- zainstalowanymi środowiskami programistycznymi,

W sali lekcyjnej powinny znajdować się e-booki, plansze dydaktyczne, schematy, grafiki i plansze interaktywne, gry edukacyjne, filmy instruktażowe (tutoriale), filmy edukacyjne, sekwencje filmowe, wizualizacje lub animacje 2D/3D, galerie zdjęć, symulatory, umożliwiające realizowanie treści w formie zdalnej i zawierające treści dotyczące:

- typów zmiennych,
- klas i obiektów,
- typowych języków programowania,

— frameworków,

Wyposażenie sali powinno obejmować między innymi zestawy instrukcji eksploatacji używanych na zajęciach urządzeń i przyrządów, zestawy ćwiczeń, instrukcje do ćwiczeń, komputerowe programy demonstracyjne i symulacyjne, czasopisma branżowe, katalogi, schematy ideowe i montażowe, normy ISO i PN, pakiety edukacyjne dla słuchaczy, karty samooceny, karty pracy dla słuchaczy i prezentacje multimedialne.

Warunki realizacji

Zajęcia powinny być prowadzone z wykorzystaniem różnych form pracy aktywizującej słuchaczy np. praca w grupach po 2-3 słuchaczy. W trakcie prac ze słuchaczami należy pozostawiać im dodatkowy czas na własne prace związane z realizowanymi celami kształcenia. Dodatkowy czas należy też poświęcić na indywidualizowanie pracy słuchaczy w zależności od ich możliwości i potrzeb. W czasie prowadzenia zajęć w pracowni należy stosować zasadę, iż nieudane ćwiczenie też może być wysoko ocenione pod warunkiem, iż uczestnik kursu potrafi wyjaśnić przyczyny niepowodzenia oraz wskazać jak powinno ono przebiegać w prawidłowy sposób. Pozwoli to na indywidualizację prowadzonych działań przez słuchaczy oraz pokaże, iż doświadczenie można zdobyć nie tylko poprzez udane doświadczenia.

4.1.5 Proponowane metody sprawdzania osiągnięć edukacyjnych słuchacza/uczestnika

Sprawdzanie opanowania przez słuchacza/uczestnika wymagań programowych będzie przeprowadzone na podstawie wykonanych ćwiczeń. W ocenie należy uwzględnić następujące kryteria ogólne: zawartość merytoryczną ćwiczeń, ich poprawność, formy przedstawienia. Sprawdzanie osiągnięć powinno odbywać się przez cały okres realizacji programu zajęć na podstawie kryteriów przedstawionych na początku kursu.

5. Ewaluacja programu kursu dodatkowych umiejętności zawodowych

Efekt kształcenia z podstawy programowej (oznaczony w programie kursu jako kluczowy dla kwalifikacji lub jednostki efektów) (czy uczeń potrafi:)	Wskaźniki potwierdzające osiągnięcie efektu kształcenia	Metody/techniki badania	Termin badania
INF.04.4. Programowanie obiektowe			
korzysta z typów danych (ek)	stosuje proste i złożone typy danych	wykład informacyjny, pokaz z objaśnieniem, wykład problemowy, film dydaktyczny, dyskusja dydaktyczna, burza mózgów, pokaz z instruktażem, pokaz z objaśnieniem,	W czasie realizacji programu nauczania podczas trwania kursu dodatkowych umiejętności zawodowych
	deklaruje własne typy danych		
	deklaruje zmienne różnych typów danych		
	wykonuje operacje na zmiennych: wejścia i wyjścia, arytmetyczne, logiczne		
	stosuje typy złożone i operacje na nich		
stosuje wyrażenia, instrukcje i biblioteki (ek)	stosuje operatory arytmetyczne, przypisania, porównania, logiczne, operatory do obsługi łańcuchów, bitowe		



Efekt kształcenia z podstawy programowej (oznaczony w programie kursu jako kluczowy dla kwalifikacji lub jednostki efektów) (czy uczeń potrafi:)	Wskaźniki potwierdzające osiągnięcie efektu kształcenia	Metody/techniki badania	Termin badania
	wykorzystuje priorytety operatorów do właściwego budowania wyrażeń stosuje instrukcję warunkową i wyboru stosuje instrukcje pętli korzysta z wybranych bibliotek języka C++, C#, Python lub innego języka programowania: biblioteka standardowa, biblioteka z funkcjami matematycznymi, biblioteka z podstawowymi algorytmami	ćwiczenia przedmiotowe, ćwiczenia produkcyjne, metoda projektów, metoda przewodniego tekstu, metoda przypadków, metoda sytuacyjna, inscenizacja, dyskusja dydaktyczna, gry dydaktyczne, teksty zamknięte, próby pracy, testy zamknięte, praca w grupie	
stosuje zasady programowania obiektowego (ek)	stosuje obiektowe podejście do rozwiązywania problemów charakteryzuje pojęcia klasa, obiekt, metoda, pole, dziedziczenie, hermetyzacja, polimorfizm dzieli zagadnienie na klasy powołuje obiekty planuje aplikację z zastosowaniem hermetyzacji, dziedziczenia i polimorfizmu		
definiuje klasy (ek)	definiuje pola klasy określa zakres widoczności pól klasy i definiuje kwalifikatory dostępu definiuje metody klasy definiuje konstruktory, w tym konstruktor kopiujący, i destruktor klasy definiuje instrukcje inicjujące konstruktora określa zakres widoczności metod klasy i definiuje kwalifikatory dostępu implementuje funkcjonalność klasy deklaruje obiekty i odwołuje się obiektem do składowych klasy definiuje składniki statyczne klasy rozdziela klasy dziedziczone i zaprzyjaźnione tworzy funkcje zaprzyjaźnione z klasą stosuje składnik statyczny klasy i metody do ich obsługi		

6. Wykaz literatury oraz niezbędnych środków i materiałów dydaktycznych

6.1. Wykaz literatury

Proponowane podręczniki:

- Kwalifikacja INF.04. Projektowanie, programowanie i testowanie aplikacji. Część 1. Inżynieria programowania - projektowanie oprogramowania, testowanie i dokumentowanie aplikacji. Podręcznik do nauki zawodu technik programista,
- Testowanie oprogramowania. Podręcznik dla początkujących, Rafał Pawlak,
- Zarządzanie projektami informatycznymi. Subiektywne spojrzenie programisty, Joel Spolsky,
- Zarządzanie projektami IT, Joseph Philips,
- Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.

Literatura dodatkowa:

- Krzysztof Szczęch, Wanda Buwała, Bezpieczeństwo i higiena pracy, Podręcznik do kształcenia zawodowego. WSiP. Warszawa 2016.
- Marcin Czerwonka, Zenon Nowocień Kwalifikacja INF.02. Administracja i eksploatacja systemów komputerowych, urządzeń peryferyjnych i lokalnych sieci komputerowych. Część 1. Systemy komputerowe. Podręcznik do nauki zawodu technik informatyk, wyd. Helion,
- Jolanta Pokorska, Kwalifikacja INF.03. Tworzenie i administrowanie stronami i aplikacjami internetowymi oraz bazami danych. Część 2. Projektowanie i administrowanie bazami danych. Podręcznik do nauki zawodu technik informatyk i technik programista, wyd. Helion,
- Jolanta Pokorska, Podręcznik do zawodu technik informatyk, technik programista, część 1, Tworzenie stron internetowych, Kwalifikacja INF.03. Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych, wyd. Helion –podręcznik w przygotowaniu.
- Adam Freeman, HTML5. Przewodnik encyklopedyczny, wyd. Helion
- David Sawyer McFarland, CSS3 nieoficjalny podręcznik, wyd. Helion
- David Sawyer McFarland, JavaScript i jQuery. Nieoficjalny podręcznik, wyd. Helion,
- Eric T. Freeman, Elisabeth Robson, Programowanie w Javascript. Rusz głową!, wyd. Helion
- Luke Welling, Laura Thomson, PHP i MySQL. Tworzenie stron WWW. Vademecum profesjonalisty, wyd. Helion
- R. Sama, K. Sama, Język angielski zawodowy w branży informatycznej, wyd. WSiP, Warszawa 2016
- Bjarne Stroustrup, Język C++. Kompendium wiedzy, wyd. Helion,
- Stephen Prata, Język C++. Szkoła programowania. Wydanie VI, wyd. Helion,
- Grębosz Jerzy, Opus magnum C++11, Programowanie w języku C++ (komplet), Wyd.: Grębosz Jerzy,
- Python dla każdego. Podstawy programowania – Michael Dawson, wyd. Helion,
- Eric Matthes, Python. Instrukcje dla programisty, wyd. Helion,
- Al Sweigart, Automatyzacja nudnych zadań z Pythonem, wyd. Helion,
- Dawn Griffiths, David Griffiths, Android. Programowanie aplikacji. Rusz głową! Wydanie II, wyd. Helion,



- Marcin Płonkowski, Android Studio. Tworzenie aplikacji mobilnych. wyd. Helion,
- Matt Neuburg, iOS 12. Wprowadzenie do programowania w Swiftie. Wydanie V, wyd. Helion,
- Steven F. Daniel, Xamarin. Tworzenie interfejsów użytkownika, wyd. Helion.
- Gniewomir Sarbicki, Python. Kurs dla nauczycieli i studentów, wyd. Helion,
- Adriaan de Jonge, Phillip Dutson, jQuery, jQuery UI oraz jQuery Mobile. Receptury, wyd. Helion,
- Jon Duckett, JavaScript i jQuery. Interaktywne strony WWW dla każdego. Podręcznik Front-End Developera wyd. Helion,
- David Herron, Platforma Node.js. Przewodnik webdevelopera. Wydanie III, wyd. Helion,
- Antonio Mele, Django 2. Praktyczne tworzenie aplikacji sieciowych. Wydanie II, wyd. Helion.

Czasopisma branżowe:

- Programista (programista.pl),

6.2. Wykaz niezbędnych środków i materiałów dydaktycznych

Pracownia programistyczna:

stanowisko komputerowe dla nauczyciela z:

- dostępem do internetu,
- pakietem programów biurowych
- projektorem multimedialnym lub tablicą multimedialną,
- urządzenie wielofunkcyjne sieciowe (drukarka z skanerem),
- środowiska programistyczne,
- monitorem minimum 24”,

Stanowiska komputerowe kursantów (1 stanowisko na 1 osobę) powinny być wyposażone następująco:

- komputer z połączeniem do internetu z zainstalowanymi środowiskami programistycznymi i ekranem minimum 21”,
- zainstalowane stosowne środowiska programistyczne,
- zainstalowane oprogramowanie biurowe,
- zainstalowane oprogramowanie do obsługi zgłoszeń i błędów,
- zainstalowane oprogramowanie do zarządzania i kontroli wersji oprogramowania,
- zainstalowane oprogramowanie wspierające testowanie aplikacji,
- zainstalowane oprogramowanie wspierające projektowanie aplikacji,
- zainstalowanymi bibliotekami i frameworkami do tworzenia aplikacji webowych,
- zainstalowanymi serwerami z obsługą PHP/Python/JS itp. i baz danych.
- zainstalowanymi bibliotekami i frameworkami do tworzenia aplikacji mobilnych,

— zainstalowane emulatory urządzeń mobilnych z system Android.

W salach lekcyjnych oraz pomieszczeniach praktycznej nauki zawodu powinny znajdować się przepisy BHP dotyczące pracy z urządzeniami oraz instrukcje obsługi i konserwacji tych urządzeń. Niezbędne są również zestawy ćwiczeń, instrukcje do ćwiczeń, pakiety edukacyjne dla słuchaczy, karty samooceny, karty pracy dla słuchaczy i prezentacje multimedialne związane z treściami kształcenia, czasopisma branżowe, katalogi, normy ISO i PN.

Dodatkowo do dyspozycji wskazane są tematyczne e-booki, animacje 2D/3D, atlasy interaktywne, plansze interaktywne, gry edukacyjne, filmy edukacyjne, symulatory, wirtualne laboratoria, programy ćwiczeniowe do projektowania przez dobieranie umożliwiające realizowanie treści w formie zdalnej dostosowane treściami do poszczególnych przedmiotów teoretycznych i praktycznych.

7. Sposób i forma zaliczenia kursu

Kurs dodatkowych umiejętności zawodowych kończy się zaliczeniem w formie walidacji osiągnięć uczestnika kursu, polegającej na ocenie wykonywanych w trakcie nauki projektów i ćwiczeń oraz na podstawie uzyskanych w trakcie kursu ocen z poszczególnych przedmiotów.

Do oceny osiągnięć edukacyjnych słuchaczy proponuje się stosowanie testów wielokrotnego wyboru, zadań z luką, ocenę aktywności słuchacza podczas wykonywania zadań w grupie, ocenę jakości wykonania zadań przez słuchacza. Proponuje się, aby osiągnięcia słuchaczy oceniać w zakresie zaplanowanych, uszczegółowionych celów kształcenia na podstawie:

- obserwacji wykonanych ćwiczeń,
- testu pisemnego.

Umiejętności praktyczne proponuje się sprawdzać na podstawie obserwacji czynności wykonywanych przez słuchacza w trakcie realizacji ćwiczeń. Podczas obserwacji należy zwrócić uwagę na:

- wyszukiwanie i przetwarzanie rzetelnych informacji pozyskanych z różnych źródeł,
- poprawność merytoryczną wykonanych ćwiczeń praktycznych,
- umiejętność pracy w zespole.

Ważne kryteria oceny efektów kształcenia to: zaplanowanie wykonania zadania, dobór elementów oraz sporządzona dokumentacja techniczna. Możliwe są również inne sposoby i formy zaliczenia, takie jak: testy praktyczne, wykonanie projektów, próby pracy, aktywność uczącego się na zajęciach, prezentacje na forum grupy z przeprowadzonych prac. Osoba, która uzyskała zaliczenie, otrzymuje zaświadczenie o ukończeniu kursu dodatkowych umiejętności zawodowych.

Osoba, która uzyskała zaliczenie, otrzymuje zaświadczenie o ukończeniu kursu umiejętności zawodowych. Osoba, która ukończy również kursy umiejętności zawodowych z jednostek efektów kształcenia:

- INF.04.1. Bezpieczeństwo i higiena pracy,
- INF.04.1. Bezpieczeństwo i higiena pracy
- INF.04.2. Podstawy informatyki
- INF.04.3. Projektowanie oprogramowania
- INF.04.5. Programowanie aplikacji desktopowych
- INF.04.6. Programowanie aplikacji mobilnych
- INF.04.7. Programowanie aplikacji zaawansowanych webowych

Program nauczania kursu umiejętności zawodowych
INF.04.4 Programowanie obiektowe

- INF.04.8. Testowanie i dokumentowanie aplikacji
- INF.04.9. Język obcy zawodowy

i otrzymała zaświadczenie o ich ukończeniu może przystąpić do egzaminu zawodowego potwierdzającego kwalifikację INF.04. Projektowanie, programowanie i testowanie aplikacji, organizowanego przez Okręgową Komisję Egzaminacyjną, po zdaniu którego otrzymuje certyfikat kwalifikacji zawodowej.

8. Sprawdzenie kompletności i poprawności opracowanego programu zajęć

Tabela 4. Tabela weryfikacji programu nauczania kursu dodatkowych umiejętności zawodowych pod kątem zgodności z przepisami prawa oświatowego

Lp.	Program kursu dodatkowych umiejętności zawodowych uwzględnia	Zawartość opracowanego programu zajęć (T/N)
1.	Cele kształcenia (zadania zawodowe)	T
2.	Efekty kształcenia	T
3.	Kryteria weryfikacji	T
4.	Warunki realizacji kształcenia w kwalifikacji (lub niezbędne do realizacji danej jednostki efektów)	T
5.	Minimalna liczba godzin kształcenia zawodowego dla kwalifikacji wyodrębnionej w zawodzie lub jednostki efektów	T

Tabela 5. Tabela weryfikacji programu kursu dodatkowych umiejętności zawodowych pod kątem kompletności efektów kształcenia

Efekty kształcenia oraz kryteria weryfikacji określone w podstawie programowej kształcenia w zawodzie		Zawartość opracowanego programu zajęć (temat zajęć)
INF.04.4. Programowanie obiektowe		
wykorzystuje środowisko programistyczne dla obiektowych aplikacji konsolowych (ew)	rozdziela kompilatory i interpretery	Środowiska programistyczne, programowanie strukturalne Programowanie z wykorzystaniem bibliotek Wstęp do programowania obiektowego
	charakteryzuje zadania kompilatora, interpretera, debuggera	Środowiska programistyczne, programowanie strukturalne Programowanie z wykorzystaniem bibliotek Wstęp do programowania obiektowego



Efekty kształcenia oraz kryteria weryfikacji określone w podstawie programowej kształcenia w zawodzie		Zawartość opracowanego programu zajęć (temat zajęć)
	analizuje błędy w kodzie za pomocą debuggera	Środowiska programistyczne, programowanie strukturalne Programowanie z wykorzystaniem bibliotek Niezawodność oprogramowania Wstęp do programowania obiektowego Programowanie obiektowe Dziedziczenie i hierarchia
	charakteryzuje etapy kompilacji i interpretacji kodu	Środowiska programistyczne, programowanie strukturalne Programowanie z wykorzystaniem bibliotek
	charakteryzuje pojęcie biblioteki	Programowanie z wykorzystaniem bibliotek Wstęp do programowania obiektowego Programowanie obiektowe
	kompiluje i uruchamia programy	Środowiska programistyczne, programowanie strukturalne Programowanie z wykorzystaniem bibliotek Niezawodność oprogramowania Wstęp do programowania obiektowego Programowanie obiektowe Dziedziczenie i hierarchia
przestrzega zasad programowania (ew)	dzieli program na funkcje (metody)	Środowiska programistyczne, programowanie strukturalne
	stosuje rekurencję	Środowiska programistyczne, programowanie strukturalne Wstęp do programowania obiektowego
	implementuje algorytmy w programie	Środowiska programistyczne, programowanie strukturalne Wstęp do programowania obiektowego
korzysta z typów danych (ek)	stosuje proste i złożone typy danych	Środowiska programistyczne, programowanie strukturalne Wstęp do programowania obiektowego



Efekty kształcenia oraz kryteria weryfikacji określone w podstawie programowej kształcenia w zawodzie		Zawartość opracowanego programu zajęć (temat zajęć)
	deklaruje własne typy danych	Środowiska programistyczne, programowanie strukturalne Wstęp do programowania obiektowego
	deklaruje zmienne różnych typów danych	Środowiska programistyczne, programowanie strukturalne Wstęp do programowania obiektowego
	wykonuje operacje na zmiennych: wejścia i wyjścia, arytmetyczne, logiczne	Środowiska programistyczne, programowanie strukturalne Wstęp do programowania obiektowego
	stosuje typy złożone i operacje na nich	Środowiska programistyczne, programowanie strukturalne Wstęp do programowania obiektowego
stosuje wyrażenia, instrukcje i biblioteki (ek)	stosuje operatory arytmetyczne, przypisania, porównania, logiczne, operatory do obsługi łańcuchów, bitowe	Środowiska programistyczne, programowanie strukturalne Programowanie z wykorzystaniem bibliotek Wstęp do programowania obiektowego
	wykorzystuje priorytety operatorów do właściwego budowania wyrażeń	Środowiska programistyczne, programowanie strukturalne Programowanie z wykorzystaniem bibliotek Wstęp do programowania obiektowego
	stosuje instrukcję warunkową i wyboru	Środowiska programistyczne, programowanie strukturalne Programowanie z wykorzystaniem bibliotek Wstęp do programowania obiektowego
	stosuje instrukcje pętli	Środowiska programistyczne, programowanie strukturalne Programowanie z wykorzystaniem bibliotek Wstęp do programowania obiektowego



Efekty kształcenia oraz kryteria weryfikacji określone w podstawie programowej kształcenia w zawodzie		Zawartość opracowanego programu zajęć (temat zajęć)
	korzysta z wybranych bibliotek języka C++, C#, Python lub innego języka programowania: biblioteka standardowa, biblioteka z funkcjami matematycznymi, biblioteka z podstawowymi algorytmami	Programowanie z wykorzystaniem bibliotek Wstęp do programowania obiektowego
stosuje zasady programowania obiektowego (ek)	stosuje obiektowe podejście do rozwiązywania problemów	Wstęp do programowania obiektowego
	charakteryzuje pojęcia klasa, obiekt, metoda, pole, dziedziczenie, hermetyzacja, polimorfizm	Wstęp do programowania obiektowego
	dzieli zagadnienie na klasy	Wstęp do programowania obiektowego
	powołuje obiekty	Wstęp do programowania obiektowego
	planuje aplikację z zastosowaniem hermetyzacji, dziedziczenia i polimorfizmu	
definiuje klasy (ek)	definiuje pola klasy	Wstęp do programowania obiektowego Programowanie obiektowe Dziedziczenie i hierarchia
	określa zakres widoczności pól klasy i definiuje kwalifikatory dostępu	Wstęp do programowania obiektowego Programowanie obiektowe Dziedziczenie i hierarchia
	definiuje metody klasy	Wstęp do programowania obiektowego Programowanie obiektowe Dziedziczenie i hierarchia
	definiuje konstruktory, w tym konstruktor kopiujący, i destruktor klasy	Wstęp do programowania obiektowego Programowanie obiektowe Dziedziczenie i hierarchia
	definiuje instrukcje inicjujące konstruktora	Wstęp do programowania obiektowego Programowanie obiektowe Dziedziczenie i hierarchia
	określa zakres widoczności metod klasy i definiuje kwalifikatory dostępu	Wstęp do programowania obiektowego Programowanie obiektowe Dziedziczenie i hierarchia



Efekty kształcenia oraz kryteria weryfikacji określone w podstawie programowej kształcenia w zawodzie		Zawartość opracowanego programu zajęć (temat zajęć)
	implementuje funkcjonalność klasy	Wstęp do programowania obiektowego Programowanie obiektowe Dziedziczenie i hierarchia
	deklaruje obiekty i odwołuje się obiektem do składowych klasy	Wstęp do programowania obiektowego Programowanie obiektowe Dziedziczenie i hierarchia
	definiuje składniki statyczne klasy	Wstęp do programowania obiektowego Programowanie obiektowe Dziedziczenie i hierarchia
	rozdziela klasy dziedziczone i zaprzyjaźnione	Dziedziczenie i hierarchia
	tworzy funkcje zaprzyjaźnione z klasą	Dziedziczenie i hierarchia
	stosuje składnik statyczny klasy i metody do ich obsługi	Dziedziczenie i hierarchia
definiuje klasy pochodne (ew)	buduje hierarchię dziedziczenia klas w programie	Dziedziczenie i hierarchia
	wydziela metody i pola do odpowiednich klas w hierarchii dziedziczenia	Dziedziczenie i hierarchia
	definiuje klasy bazowe i pochodne	Dziedziczenie i hierarchia
	stosuje metody wirtualne, definiuje klasy abstrakcyjne	Dziedziczenie i hierarchia
programuje szablony (wzorce) klas (ew)	definiuje szablony klas dla obsługi prostych typów liczbowych	Dziedziczenie i hierarchia
programuje obsługę wyjątków (ew)	stosuje szkielet obsługi wyjątków z instrukcjami try i catch	Niezawodność oprogramowania
	stosuje instrukcję throw	Niezawodność oprogramowania
	opracowuje listę możliwych błędów wykonania aplikacji	Niezawodność oprogramowania
	definiuje obsługę dla błędów wykonania aplikacji	Niezawodność oprogramowania