프로그래밍 정답

1	⑤	2	②	3	⑤	4	①	5	②
6	②	7	①	8	③	9	①	10	③
11	③	12	②	13	④	14	④	15	④
16	⑤	17	②	18	④	19	③	20	④

해 설

1. **[출제의도] 프로그래밍 언어의 계층 구조와 기계어에 대해 설명할 수 있다.**
   기계어는 '0'과 '1'의 2진수만을 사용하여 명령어와 데이터를 나타내며, 컴퓨터가 직접 인식하여 실행할 수 있는 언어이다.
   [오답풀이] ②번은 인터프리터 언어에 대한 설명이고, ③번은 고급 언어에 대한 설명이며, ④번은 C 언어나 어셈블리 언어에 대한 설명이다.
2. **[출제의도] 프로그래밍 언어의 개념과 특징을 설명할 수 있다.**
   프로그래밍 언어는 크게 저급언어와 고급언어로 분류할 수 있으며 어셈블리 언어는 저급언어에 속한다. 프로그래밍 언어는 비슷한 부분을 많이 가지고 있어 하나의 프로그래밍 언어를 잘 알고 있으면 다른 언어의 학습에 도움이 된다.
3. **[출제의도] 알고리즘의 의미를 이해하고 설명할 수 있다.**
   알고리즘은 어떤 문제를 해결하기 위한 논리적인 절차를 말하는 것으로, 이 알고리즘을 알맞게 구성하면 복잡한 문제도 체계적, 효율적으로 해결할 수 있다. 알고리즘은 기본적인 몇 단계를 거쳐 일정한 횟수를 반복 수행하면 반드시 결론에 도달한다.
   [오답풀이] 알고리즘은 C 언어로도 많이 작성한다.
4. **[출제의도] 프로그램의 작성 절차를 설명할 수 있다.**
   프로그램의 작성 절차는 주어진 문제 분석, 입출력의 종류와 형식 설계, 알고리즘이나 순서도 작성과 이를 기초로 한 프로그램 코딩 및 실행 과정을 거쳐 잘못된 부분을 수정하여 완성한다. 따라서 (다)의 과정에는 ② 프로그램의 작성과 입력, ③ 프로그램 실행, ④ 프로그램의 번역 및 오류 수정 작업, ⑤ 프로그램의 모의 실행 부분이 포함된다.
   [오답풀이] 입출력 장치는 (나) 단계의 프로그램 설계 단계에서 결정한다.
5. **[출제의도] 객체지향 언어의 특징을 알고, 객체지향 언어의 종류를 설명할 수 있다.**
   절차지향 언어는 "어떻게 처리하느냐"에 주안점을 둔 언어고 객체지향 언어는 "무엇을 처리하느냐"에 주안점을 둔 언어이다. 객제지향 언어는 객체와 클래스를 이용하여, 마우스 클릭 등과 같은 이벤트 처리에 편리하다. 객체지향 언어로는 비주얼 C++, 자바 등이 있다.
6. **[출제의도] 순서도를 해독하고 작성할 수 있다.**
   SW 가 0일 경우는 변수 K 에 A(홀수)를 더하고, SW가 1일 경우는 변수 M에 A(짝수)를 더한다. SW는 0과 1로 변하면서 마치 스위치와 같은 역할을 한다.
7. **[출제의도] 문제를 파악하고 알고리즘을 이해하며 순서도를 구현할 수 있다.**
   ㉠은 10으로 나누면서 자리수를 줄여 나가는 방법을 이용하고 있으며, ㉡에는 y의 값을 x로 옮기는 과정이 필요하다. 이 문제에서 중요한 점은 x에 소수점이 절삭된 정수 몫이 들어가야 된다는 것이다.
8. **[출제의도] 변수의 개념과 자료형을 이해하고 실제 문제에 적용 할 수 있다.**
   프로그램의 시작부터 끝까지 변하지 않는 값은 상수로 선언할 수 있으며 소수 이하자리의 연산이나 큰 값이 예상되는 변수는 정확성을 위해 정수형보다는 실수형 변수가 사용된다.
   [오답풀이] 행성의 적도 둘레를 구할 때 필요한 연산자는 산술 연산자이며 논리 연산자는 필요하지 않다.
9. **[출제의도] 여러 가지 연산자의 의미를 알고 바르게 사용할 수 있다.**
   주어진 조건을 충족하기 위해서는 평균(Avg)이 60 이상이어야 하고, 국어(Kor) 점수가 40 이상이어야 한다는 두 개의 조건이 모두 참이어야 한다. 따라서 두 조건은 AND(&&) 연산자로 연결되어야 한다.
10. **[출제의도] 자료형이란 무엇인지 이해하고 자료형의 종류 및 특징을 설명할 수 있다.**
    상수(constant)는 값이 한번 정의되면 프로그램이 실행되는 도중에 변하지 않는다. 변수(variable)는 상수와는 달리 프로그램이 실행되는 도중에 그 값이 변할 수 있다. 보통 변수를 사용할 때는 그 데이터형을 선언한다. 주석문은 프로그램을 설명할 때 사용하는 것으로 프로그램의 실행에는 영향을 주지 않는다.
11. **[출제의도] 여러 가지 연산자의 의미를 알고 이들의 우선 순위에 대하여 설명할 수 있다.**
    산술 연산자의 우선 순위 : ( ) → *, / → +,- 의 순으로 연산 진행.
    관계 연산자의 우선 순위 : >,<,>=,<= → !=,== 의 순으로 연산 진행.
    논리 연산자의 우선 순위 : != → && → || 의 순으로 연산 진행.
    여러 종류의 연산자가 혼용되었을 때는 산술 연산 → 관계연산 → 논리연산의 순으로 연산이 진행된다.
12. **[출제의도] 반복문의 동작 방식을 설명할 수 있다.**
    반복문은 조건이 만족 될 때까지 반복 수행하는 기능을 가진다.
    [오답풀이] if(If)문은 조건은 처리할 수 있으나 반복 기능이 없다. goto(Goto) 문의 분기 기능을 이용하여 반복 처리를 할 수도 있으나 goto(Goto)문 만으로는 조건 처리를 할 수 없다.
13. **[출제의도] 반복문의 동작 방식과 증가 값 기능을 설명할 수 있다.**
    일반적으로 for(For)문에서는 1을 증가 값으로 많이 사용하나 본 문제에서는 3을 이용하고 있으므로 for(For)문 안의 문장은 2번 수행된다.
14. **[출제의도] 기본 명령문인 switch(Visual Basic에서는 Select Case)문을 이용하여 프로그램을 작성할 수 있다.**
    a가 23이므로 b=a % 5의 값은 3이 된다. 따라서 default(Visual Basic에서는 Case Else)문을 수행하므로 'D'를 출력한다.
15. **[출제의도] 이중 반복문의 사용법을 알고 활용할 수 있다.**
    a=1일 때, * a=2일 때, **
    a=3일 때, *** a=4일 때, ****
    a=5일 때, *****
16. **[출제의도] 알고리즘의 의미를 알고, 효과적인 알고리즘을 적용할 수 있다.**
    이 프로그램은 1 + (1+2) + (1+2+3) + ... + (1+2+...+9+10)을 계산하는 프로그램이다. (가) 프로그램에서는 1부터 b까지의 합을 새롭게 계산하여 사용하지만, (나) 프로그램은 1부터 b까지의 합이 1부터 b-1까지의 합에 b를 더하면 된다는 것을 이용하여, 불필요한 반복 동작을 하지 않도록 개선하고 있다.
17. **[출제의도] 조건문의 사용법을 알고, 바르게 사용할 수 있다.**
    if(조건) 문에서 ( ) 안의 조건이 성립하면 if 다음의 명령을 실행하고, 성립하지 않으면 else 다음의 명령을 실행한다.
18. **[출제의도] 반복문을 사용하여 프로그램을 작성할 수 있다.**
    Do Loop While 문은 while문 내의 명령을 무조건 한 번 실행한 후 조건을 판단한다. 따라서 문제에서 조건이 a<10이므로 a가 5, 7, 9인 경우에 수행하게 된다.
19. **[출제의도] 프로그램의 기본 명령문을 해독하고 반복문의 동작방식과 특징을 설명할 수 있다.**
    do while(Do Loop While)문은 조건 검사전에 먼저 실행되고, for(FOR)문은 증가값을 더한 뒤 종료 조건과 비교하기 때문에 증가 값은 for(FOR)문이 정상적으로 시행되면 종료값 보다 많게 된다. 본 문제에서는 (가)와 (나)의 출력값이 6이다.
20. **[출제의도] 프로그램에서 발생하는 각종 오류를 구별하고 오류가 발생하는 원인을 찾아 낼 수 있다.**
    본 문제에서 문법오류는 없으나 ㉤에서 0으로 나눗셈을 시행하는 논리(연산)오류를 범하고 있다.
    [오답풀이] ㉡은 if(If)문의 조건절에 조건문이 없기 때문에 오류로 생각하기 쉬우나 변수의 값으로 참 거짓을 판단할 수 있다. 즉, 변수가 가지고 있는 값이 1이상이면 참 0이면 거짓이 된다.