

평가지
<div>1. “Hello java”를 출력하는 프로그램이다. 작성한 자바소스파일을 저장하고 컴파일하고 실행하시오</div> <div><pre>public class HelloJava{ public static void main(String[] args) { System.out.println(" Hello java"); } }</pre></div> <div>(1) 이 자바소스파일을 저장한 소스파일 이름은 무엇인가 선택하시오? <div>① javac HelloJava ② HelloJava.java ③ HelloJava ④ java HelloJava</div></div> <div>(2) 이 자바 소스파일을 컴파일하는 명령어가 올바른 것을 선택하시오 <div>① javac HelloJava.java ② HelloJava.java ③ HelloJava.class ④ java HelloJava</div></div> <div>(3) 위에서 만들어진 클래스파일을 실행하는 자바 명령어를 선택하시오 <div>① javac HelloJava.java ② HelloJava.java ③ java HelloJava ④ HelloJava.class</div></div> <div>2. 배열을 선언하고 생성하는 다음 물음에 답하라.</div> <div>(1) 10 개의 문자를 저장할 수 있는 배열 arr 를 생성하는 코드를 작성한 코드로 알맞은 것은? <div>① int[] arr= new int[10] ; ② double[] arr = new double[10]; ③ char[] arr= new char{10}; ④ String[] arr = new String[10];</div></div>

(2) “apple” , “banana” , “tomato”를 초기값으로 갖는 문자열 배열을 선언하시오

① String[] arr= { “apple” ,“banana” ,“tomato”};
② char[] arr= { “apple” ,“banana” ,“tomato”};
③ double[] arr= { “apple” ,“banana” ,“tomato”};
④ int[] arr= { “apple” ,“banana” ,“tomato”};

(3) 이름을 5개 입력받아 문자열 배열에 저장하는 프로그램의 일부입니다.
코드의 일부를 완성하시오, name배열의 요소에 이름을 저장하는 코드입니다. 알맞은 코드를 선택 하세요

```
public class ArrayTest  
{  
    public static void main(String[] args)  
    {  
        String[ ] name = new String[5];  
        Scanner sc = new Scanner(System.in);  
  
        for(int i=0; i<name.length; i++)  
        {  
            System.out.println("이름을입력:" );  
            String tmp= sc.next( );  
  
        }  
    }  
}
```

① name[0] = tmp;
② name[i] = tmp;
③ name[1] = tmp;
④ name[2] = tmp;

답)

3. 물음에 답하시오

(1) 다음 코드에 중 오류가 있는 것은 ?
①Score[] p = new Score[10]; ② Score p [] = new Score[10];
③Score p[10] = new Score[10]; ④ Score[] p =null;

(2) 생성자에 대한 설명 중 틀린 것은 ?
① 생성자의 이름은 클래스의 이름과 반드시 같아야 한다.
② 상속시 자식의 생성자가 호출될 때 부모의 기본생성자가 기본적으로 호출된다.
③ 생성자에서는 아무 값도 리턴하지 않기 때문에 리턴타입을 void로 해 주어야 한다.
④ 생성자가 작성되지 않으면 컴파일러가 자동으로 기본 생성자를 추가해 준다.

답

4. 다음 중 메서드 오버로딩(클래스내에 이름이 같은 메서드가 여러개 존재하는 것) 에 실패한 사례는 ?

① class A{
int x;
void method(int a) { System.out.println(a); }
int method(int b) { return x+b; }
}

② class A{
int x;
void method(int a) {System.out.println(a); }
void method() { System.out.println("호출") ; }
}

③ class A{
int x;
int method() {return 5;}
int method(int b) {return x+b;}
}

④ class A{
int x;
int method(int a) {return a+a;}
int method() {return 5;}
}

5. 다음의 상속관계에서 오류가 발생하는 코드를 고르시오

```
class Person{  
    public String name;  
    public int age;  
}
```

```
public class Doctor extends Person{  
    public void 치료하다(){  
        System.out.println("치료하다");  
    }  
}  
public class Nurse extends Person{  
    public void 간호하다(){  
        System.out.println("간호하다");  
    }  
}
```

① Person p = new Doctor() ;
③ Nurse o = new Doctor() ;

② Nurse n = new Nurse();
④ Object o = new Doctor();

답)

(2) 다음의 소스에 틀린 부분이 있는지 판단하라.

Calculator클래스의 add 메소드를 사용하려고 한다. 무엇이 잘못된가?

올바르게 수정된 main메서드 내부의 코드를 선택하시오

```
class Calculator{  
    public int add( int a, int b) {  
        return a+b;  
    }  
}
```

```
class Program{  
    public static void main(String[] args){  
        int result = add( 4,8);  
        System.out.println( "합계=" + result );  
    }  
}
```

①public static void main(String[] args)
{
 Calculator t = new Calculator();
 int sum = t.add(2,4);
}

②public static void main(String[] args)
{
 int sum = Calculator.add(2,4);
}

③public static void main(String[] args)
{
 Calculator t = new Calculator();
 double sum = t.add(2,4);
}

④public static void main(String[] args)
{
 char sum = add(2,4);
}

6. Score 타입의 객체만을 저장할 수 있는 ArrayList 컬렉션의 list 객체를 생성 하는 코드를 아래의 보기에서 선택하시오

- ① ArrayList<Object> list= new ArrayList<>();
- ② ArrayList list = new ArrayList();
- ③ ArrayList<Student> list = new ArrayList<>();
- ④ ArrayList<Score> list= new ArrayList<>();

답)

7. 다음은 생성자를 작성한 코드이다. 생성자가 잘못 작성된 코드를 선택하시오

①public class Person
{
 private String name;
 private int age;

 public Person()
 {

 }
}

②public class Person
{
 private String name;
 private int age;

 public Person(String name, int age)
 {
 this.name = name;
 this.age =age;
 }
}

③public class Person
{
 private String name;
 private int age;

 public void Person(String name)
 {
 this.name = name;
 }
}

④ public class Person
{
 private String name;
 private int age;

 public Person(int age)
 {
 this.age = age;
 }
}

답)

8. 다음 프로그램의 적절한 코드를 아래의 보기에서 선택하시오

(1) 1에서 100까지의 합을 더하는 while문이다. 빈칸(괄호 안)에 적절한 코드를 삽입하라.
int sum=0, i=1;
while(true)
{
 if()break;
 sum += i;
 i++;
}

- ① i > 100 ② i >= 100 ③ i < 99 ④ i <= 99

답

(2) 다음은 프로그램 코드의 일부이다 출력결과를 아래의 보기에서 선택하시오

```
for(int i=1 ; i<10 ; i++)  
{  
    if( i%4 ==0)  
        continue;  
  
    System.out.print(i+ " " ) ;  
}
```

- ① 1 2 3 5 6 7 9
- ② 1 2 3 4 5 6 7 8 9
- ③ 1 2 3
- ④ 1 2 3 4 5

답

(3) 다음은 프로그램 코드의 일부분이다.

입력받은 문자열이 “그만”이면 프로그램이 종료되도록 괄호 안의 코드를 작성 해 보시오

```
Scanner sc = new Scanner(System.in);
while(true)
{
    String input = sc.next();
    if(                )
        return;
}
```

- ① input.length();
- ② input.charAt(2);
- ③ input.substring(3);
- ④ input.equals(“그만”)

답)

(4) 다음은 프로그램 코드의 일부분이다.

“5” 문자열을 정수값으로 변환하여 제곱값을 출력 하려고 한다. 적절한 클래스의 메소드를 활용하여 “5”를 정수 5로 변환하는 코드이다. 올바른 것을 선택하시오

```
String number=“5”;
int n = (                );
System.out.println( “제곱값은= ” + n*n);
```

- ① Math.abs(number);
- ② Integer.sqrt(number);
- ③ Integer.parseDouble(number);
- ④ Integer.parseInt(number);

9. 다음의 Person 이라는 super class를 상속받은 Child라는 sub를 클래스를 작성하였다. 부모의 클래스를 상속 받아 speak()메서드를 오버라이딩하였다.
다음 MainEx클래스의 실행결과를 보기에서 선택하시오.

```
class Person()
{
    private String name;
    private int age;
    public void speak()
    {
        System.out.println(“사람이 말한다”);
    }
}
```

```
public class MainEx{
    public static void main(String[] args)
    {
        Person s= new Child();
        s.speak();
    }
}
```

```
class Child extends Person
{
    @Override
    public void speak()
    {
        System.out.println(“어린이가 말한다”);
    }
}
```

- ① “사람이 말한다”
- ② “어린이가 말한다”
- ③ “사람이 말한다 어린이가 말한다”
- ④ “어린이가 말한다 사람이 말한다”

답)

10. (1) 객체지향의 3대 요소에 대해서 쓰고 간단히 설명하시오

<부분점수 있으니 아는대로 서술하시오>

(2) 다음의 PolyEx class에서 Person형으로 Teacher객체를 생성하는 코드를 작성하시오

<상속관계에서는 부모형참조변수에 저장하는 업캐스팅이 가능하다 >

```
public class PolyEx{
    public static void main(String[] args)
    {
        
    }
}
```

```
class Person{
    public void work(){
        System.out.println(“사람은 일한다.”);
    }
}
```

- ① Person s ;
- ② Teacher s = new Teacher();
- ③ Teacher s ;
- ④ Person s = new Teacher();

```
class Teacher extends Person{
    @Override
    public void work(){
        System.out.println(“가르친다”);
    }
}
```

답)

작은거 하나부터 스스로 완성해 보세요^^ 더 크고 복잡한 것도 완성할 수 있게 된답니다.

된다라는 생각으로 별거 아니다라는 생각으로 쉬운 것부터 시작해 보세요. 좋은 프로그래머가 될 수 있습니다.
행운을 빕니다 😊