

問1 演算子の動作に関する次の各設問に答えよ。（20点）

設問1 変数を次のように宣言、初期化した。（各1点）

int ans, a=10, b=3;

次の各文それぞれを実行した後の変数 ans の値を答えよ。

- (1) ans = a + b;
- (2) ans = a - b;
- (3) ans = a * b;
- (4) ans = a / b;
- (5) ans = a % b;

答(1) 13 (2) 7 (3) 30 (4) 3 (5) 1

設問2 変数を次のように宣言、初期化した。（各1点）

int a=7, b=3, c=2;

次の各文それぞれを実行した後の変数 a の値を答えよ。

- (1) a += 5;
- (2) a -= 2;
- (3) a += a + 3;
- (4) a *= a;
- (5) a -= b += c;

答(1) 12 (2) 5 (3) 17 (4) 49 (5) 2

設問3 変数を次のように宣言、初期化した。（各1点）

int ans=5, a=1, b=1;

次の各文それぞれを実行した後の変数 ans の値を答えよ。

- (1) ans = ++a;
- (2) ans = a--;
- (3) ans+= --a;
- (4) ans*= a++;
- (5) ans = ++a - b++;

答(1) 2 (2) 1 (3) 5 (4) 5 (5) 1

設問4 変数を次のように宣言、初期化した。（各1点）

int a=3, b=5;

次の各文それぞれを実行した画面出力(true/false)を答えよ。

- (1) System.out.println(a > b);
- (2) System.out.println(a*2 >= b+1);
- (3) System.out.println(a != b*2);
- (4) System.out.println(a/b == 0);
- (5) System.out.println(a!=b == a<=b);

答(1) false (2) true (3) false (4) true (5) true

問2 Java言語文法に関する次の設間に答えよ。（20点）

設問1 次のコード中の誤り箇所を2つ指摘せよ。（各2点）

```
int a==2;
if(a=10)
{
    System.out.println(a);
}
```

答(1) 1行目：代入になつてない
(2) 2行目：if文の条件がboolean型になつてない設問2 次のコード中の誤り箇所を2つ指摘せよ。（各2点）

```
int a, b, c
a=10;
20=b;
c=a*b;
```

答(1) 1行目：行末にセミコロンがない
(2) 3行目：代入演算子の左右のペンドが逆である設問3 次のコード中の誤り箇所を2つ指摘せよ。（各2点）

```
int a=1;
if(A<10)
{
    System.out.println(エラーです);
}
```

答(1) 2行目：Javaでは大・小文字は区別される。このため
変数Aは宣言されておらず使用できない
(2) 4行目：文字列リテラル””になつてない設問4 次のコード中の誤り箇所を2つ指摘せよ。（各2点）

```
int a=5, b=2, c=4;
if(a<b<c)
{
    System.out.println('最大は'+c);
}
```

答(1) 2行目：演算はその優先順位に従う。このため
比較できない値同士を比較することになる
(2) 4行目：文字列リテラル””になつてない設問5 次のコード中の誤り箇所を2つ指摘せよ。（各2点）

```
double if=2.5;
int ans;if*2;
```

答(1) 1行目：変数の識別子にキーワードifを用いている。
(2) 2行目：double型からint型へ型のランクが落ちている
にもかかわらずそのまま代入を行っている。

問3 次の各コードを実行した画面出力を正確に答えよ。（25点）

- (1) System.out.print("A="+1+2);
- (2) System.out.print("B="+2*3);
- (3) System.out.print("C="+-(3-4));
- (4) System.out.print("今日は¥nJavaの中間試験¥nです");
- (5) System.out.print("努力で"+0x46+"点、運で"+0144+"点");

答(1) A=12

答(2) B=6

答(3) C=-1

答(4) 今日は
Javaの中間試験
です

答(5) 努力で70点、運で100点

問4 次の各処理を実行するコードを完成させよ。（20点）

設問1 円の面積を求めよ。

```
int radius=2; // 半径
double PI=3.14;
double area=[(1)];
```

答(1) PI*radius*radius

設問2 三角形の面積を求めよ。

```
int base=7, height=3; // 底辺と高さ
double area=[(2)];
```

答(2) (double)base*height/2 又は base*height/2.0

情報工学部	情報工学科		年次		組	学籍番号		氏名	
-------	-------	--	----	--	---	------	--	----	--

設問3 3つの整数における平均を求めよ。

```
int a=3, b=7, c=4;  
double ave= ;
```

答(3) $(\text{double})(\text{a}+\text{b}+\text{c})/3$ 又は $(\text{a}+\text{b}+\text{c})/3.0$

設問4 2点(x1,y1)、(x2,y2)を通る直線の傾きを求めよ。

```
int x1=2, y1=3, x2=4, y2=4;  
double gradient= ;
```

答(4) $(\text{double})(\text{y}_2-\text{y}_1)/(\text{x}_2-\text{x}_1)$ ※ $\text{x}_1 \neq \text{x}_2$ の場合

問5 次はある整数が偶数であるか奇数であるかを判定するコードの実行画面である。下にあるコードの空欄を適切に埋めてコードを完成させよ。(15点)

【実行画面】

```
>java Intermediate5  
整数を入力してください。  
3  
奇数です  
-- Press any key to exit (Input "c" to continue) --  
  
>java Intermediate5  
整数を入力してください。  
12  
偶数です  
-- Press any key to exit (Input "c" to continue) --
```

【考え方】

偶数 → 2で割り切れる
奇数 → 2で割り切れない

【コード】

```
import java.io.*;  
class Intermediate5  
{  
    public static void main(String[] args) throws IOException  
    {  
        // キーボード準備  
        BufferedReader br;  
        br = new BufferedReader(new InputStreamReader(System.in));  
  
        int a; // 整数  
  
        // 整数入力  
        System.out.println("整数を入力してください。");  
        a = Integer.parseInt(br.readLine());  
  
        // 偶数・奇数の判定  
        if(a%2==0)  
        {  
            System.out.println("偶数です");  
        }  
        else  
        {  
            System.out.println("奇数です");  
        }  
    }  
}
```

《参考資料》

演算子の優先順位表

主な演算子の優先順位

演算子	名前	結合規則
<code>++</code>	後置インクリメント	左
<code>--</code>	後置デクリメント	左
<code>!</code>	論理否定	右
<code>~</code>	1の補数 (反転)	右
<code>+</code>	プラス	右
<code>-</code>	マイナス	右
<code>++</code>	前置インクリメント	右
<code>--</code>	前置デクリメント	右
<code>()</code>	キャスト	右
<code>*</code>	乗算	左
<code>/</code>	除算	左
<code>%</code>	剰余	左
<code>+</code>	加算 (文字列連結)	左
<code>-</code>	減算	左
<code><<</code>	左シフト	左
<code>>></code>	右シフト	左
<code>>>></code>	符号なし右シフト	左
<code>></code>	より大きい	左
<code>>=</code>	以上	左
<code><</code>	未満	左
<code><=</code>	以下	左
<code>==</code>	等価	左
<code>!=</code>	非等価	左
<code>&</code>	ビット論理積	左
<code>^</code>	ビット排他的論理和	左
<code> </code>	ビット論理和	左
<code>&&</code>	論理積	左
<code> </code>	論理和	左
<code>? :</code>	条件	右
<code>=</code>	代入	右
<code>+=, -=など</code>	複合代入演算	右

↑ 優先度高い

↓ 同じ優先度

↓ 優先度低い

お疲れ様でした。

情報工学部

情報工学科

年次

組

学籍
番号

氏名