

EKSAMEN

Emnekode: ITF10212	Emne: Innføring i programmering
Dato: 4. desember 2012	Eksamenstid: kl 09.00 til kl 13.00
Hjelpemidler: To A4-ark (4 sider) med egenproduserte notater (håndskrevne/maskinskrevne) .	Faglærer: Tord Akerbæk
Eksamensoppgaven: <p>Oppgavesettet består av 8 sider inklusiv denne forsiden. Kontroller at oppgaven er komplett før du begynner å besvare spørsmålene.</p> <p>Oppgavesettet består av seks oppgaver med i alt 20 deloppgaver. På hver oppgave er angitt hvor mye denne teller av totalen. Karakteren settes likevel ut fra en helhetsvurdering av besvarelsen.</p> <p>Les gjennom alle oppgavene før du begynner. Oppgavene er ikke sortert etter vanskelighetsgrad.</p>	
Sensurdato: Torsdag 14. januar Karakterene er tilgjengelige for studenter på studentweb senest 2 virkedager etter oppgitt sensurfrist. Følg instruksjoner gitt på: www.hiof.no/studentweb	

Oppgave 1 Kodeforståelse (20%)

Hva blir utskriften fra disse 10 programsekvensene?

A

```
$enhet = 'gram';
$mengde = 24;
$ingrediens = 'sukker';
echo "Tilsett $mengde $enhet $ingrediens!";
```

B

```
$hovedstad = array(
    'Norge'=>'Oslo', 'Tyskland'=>'Berlin',
    'Spania'=>'Madrid', 'Nederland'=>'Amsterdam');
$land = 'Spania';
echo 'Hovedstaden i ', $land, ' heter ', $hovedstad[$land];
```

C

```
$retning = array(
    0=>'N', 45=>'NO', 90=>'O', 135=>'SO',
    180=>'S', 225=>'SV', 270=>'V', 315=>'NV');
$grad = (135 + 90) % 360;
echo "Styr mot $retning[$grad]";
```

D

```
$a = 16 % 6;
if($a > 3)
{
    if(!$a-4)
    {
        echo 'Ole ';
    }
}
elseif($a < 3)
{
    echo 'Dole ';
}
else
{
    echo 'Doffen ';
}
echo 'Duck';
```

E

```
$a = 100; $b = 100;
if(((($a >= $b) != ($a <= $b)) == (($a >= $b) xor ($a <= $b)))
    echo $a + $b;
else
    echo $a . $b;
```

F

```
$a = array(1,2,4,3,5,2,8);
$i = 0;
while($a[$i] < $a[$i+1])
{
    echo $a[$i], ' ';
    $i = $i + 1;
}
echo $a[$i];
```

G

```
for($i=0;$i<6;$i=$i+1)
{
    $p = $i * $i;
    echo $p, ' ';
}
```

H

```
function n($a)
{
    foreach($a as $i)
    {
        if($i < 0)
        {
            return $i;
        }
    }
    return 0;
}
$stall = array(100,10,1,-1,-10);
echo n($stall);
```

I

```
function tripler($s)
{
    while(strlen($s) > 0)
    {
        $r[] = substr($s,0,3);
        $s = substr($s,3);
    }
    return $r;
}
$d = tripler('Nesoddtangen');
echo $d[1];
```

J

```
$bein = array(
    'orm'=>0, 'struts'=>2, 'hund'=>4, 'maur'=>6, 'edderkopp'=>8);
function tell($b,$d)
{
    echo "\nEn $d har $b[$d] føtter.";
    return $b[$d];
}
$dyr = 'hund';
echo "\nHvor mange føtter har en $dyr?";
$svar = tell($bein,$dyr);
echo "\nSvaret er $svar.";
```

Oppgave 2 Språkkunnskap (16%)

A

Forklar forskjellen på bruken av enkle og doble anførselstegn i PHP. Gi minst 2 ulike eksempler på tekst som vil bli tolket forskjellig innenfor enkle og doble anførselstegn.

B

Se på programkoden under. Hvor mange ganger kalles funksjonen checkTemperature? Hvilken egenskap ved operatorene 'or' og 'and' er det som har innvirkning på dette?

```
function checkTemperature($label,$temp)
{
    if($temp >= 100)
    {
        return TRUE;
    }
    else
    {
        return FALSE;
    }
}
if(checkTemperature('Heater1',90) or checkTemperature('Heater2',105))
    echo "Alarm!";

if(checkTemperature('Heater3',101)and checkTemperature('Heater4',105))
    echo "Alarm, alarm!";
```

Oppgave 3 Grunnleggende programmering (18%)

Skriv følgende funksjoner.

A

En funksjon f2c som konverterer fra Fahrenheit til Celcius. Grader Fahrenheit skal være argument, grader celcius skal være returverdi.

(Bruk formelen $^{\circ}\text{C} = (^{\circ}\text{F} - 32) \times 5/9$)

B

En funksjon som tester at funksjonen f2c () fungerer med en spesifisert testverdi.

Funksjonen skal kalle f2c en gang, returnere TRUE hvis returverdien stemmer med fasit og ellers FALSE.

```
function testF2C($fahr,$fasit)
```

C

En funksjon som tester funksjonen f2c ved å kalle testF2c () 5 ganger med ulike testverdier. Hvis funksjonen feiler, skal testen avbrytes og en feilmelding skrives ut sammen med den aktuelle testverdien. Hvis alle testene går bra, skal en friskmelding skrives til slutt.

```
function fullTestF2c()
```

Oppgave 4 Datastrukturer (24%)

En tabell inneholder informasjon om studentenes resultater på 4 oppgaver.

Navn	O1	O2	O3	O4	Sluttresultat
Arne Mo	5.0	4.0	2.0	3.0	Godkjent
Berit Lie	8.0	4.0	6.0	2.0	Godkjent
Dina Vang	0.0	1.0	1.5	1.5	Ikke godkjent
Eldar Aas	3.5	3.5	2.0	1.0	Godkjent

osv. ...

I et program er tabellen representert av en todimensjonal array. Den ytre arrayen inneholder en rad for hver student. Antallet kan variere. Hver rad er i sin tur også en array. Første og siste felt (eller kolonne) i raden er tekststrenger. De fire feltene i midten er desimaltall og inneholder poengsummer for de fire oppgavene.

```
$stud = array(  
    array('Arne Mo', 5.0, 4.0, 2.0, 3.0, 'Godkjent'),  
    array('Berit Lie', 8.0, 4.0, 6.0, 2.0, 'Godkjent'),  
    array('Dina Vang', 0.0, 1.0, 1.5, 1.5, 'Ikke godkjent'),  
    array('Eldar Aas', 3.5, 3.5, 2.0, 1.0, 'Godkjent')  
    osv ...  
);
```

I dette eksemplet vil \$stab[0] inneholde Arne Mos resultater. \$stab[2][0] vil være lik 'Dina Vang' og \$stab[1][4] vil være 2.0.

A

Lag en funksjon som skriver en slik tabell som html. Navnene og overskriftene er th-tagger, det øvrige innholdet er td-tagger. Argumentet til funksjonen er den todimensjonale arrayen beskrevet over.

```
function printTable($stab)
```

B

Vi trenger en funksjon som regner gjennomsnittlig poengsum for en student. Funksjonen skal ta den indre, endimensjonale arrayen som argument, og returnere gjennomsnittet av poengsummen for de fire oppgavene.

```
function averageFromStudent($row)
```

C

Vi ønsker å finne den dyktigste studenten. Skriv en funksjon som tar den 2-dimensjonale tabellen som argument og returnerer navnet på den studenten som har høyest gjennomsnittlig poengsum.

5 Fil- og streghåndtering (10%)

Innholdet i tabellen ligger nå på fil med formatet

```
Arne Mo; 5.0;4.0;2.0;3.0;Godkjent  
Berit Lie;8.0;4.0;6.0;2.0;Godkjent  
osv.
```

Vi ønsker nok en gang å regne ut gjennomsnittlig poengsum for en student, men denne gangen skal vi hente tallene fra fila. Funksjonen skal ta filnavnet og navnet på en student som argumenter og returnere gjennomsnittlig poengsum for denne studenten. Skriv funksjonen.

```
function averageFromFile($filename, $name)
```

Tips: Hvis du synes det er hensiktsmessig, kan du bruke `averageFromStudent` som hjelpefunksjon. Dette kan du gjøre selv om du ikke har klart oppgave 4B

6 Databaser (8%)

Arrayen ligger nå i en databasetabell. Informasjonen vi trenger for å kople oss til databasen ligger i fire konstanter: `SERVER`, `USER`, `PASSWORD` og `DATABASE`.

I databasen ligger tabellen 'resultater'. Tabellen har et format som tilsvarer arrayen fra oppgave 4: Første og siste kolonne er tekststrenger, de fire kolonnene i midten har type `DECIMAL`.

Lag en ny funksjon for å regne ut gjennomsnittlig poengsum for en student. Også denne funksjonen skal returnere gjennomsnittlig poengsum for en student. Argumentet er navnet på studenten.

```
function averageFromSQL($name)
```

Også her kan du bruke `averageFromStudent` som hjelpefunksjon.

Vedlegg - PHP funksjoner

Her følger en forenklet utgave av noen manualsider.

explode — Split a string by string

Description

```
array explode ( string $delimiter , string $string )
```

Returns an array of strings, each of which is a substring of string formed by splitting it on boundaries formed by the string delimiter.

Parameters

```
delimiter
```

The boundary string.

```
string
```

The input string.

Return Values

Returns an array of strings created by splitting the string parameter on boundaries formed by the delimiter.

strlen — Get string length

Description

```
int strlen ( string $string )
```

Returns the length of the given string.

Parameters

```
string
```

The string being measured for length.

Return Values

The length of the string on success, and 0 if the string is empty.

substr — Return part of a string

Description

```
string substr ( string $string , int $start [, int $length ] )
```

Returns the portion of string specified by the start and length parameters.

Parameters

```
string
```

The input string. Must be one character or longer.

```
start
```

If start is non-negative, the returned string will start at the start'th position in string, counting from zero. For instance, in the string 'abcdef', the character at position 0 is 'a', the character at position 2 is 'c', and so forth.

If start is negative, the returned string will start at the start'th character from the end of string.

If string is less than or equal to start characters long, FALSE will be returned.

```
length
```

If length is given and is positive, the string returned will contain at most length characters beginning from start (depending on the length of string).

If length is omitted, the substring starting from start until the end of the string will be returned.

Return Values

Returns the extracted part of string; or FALSE on failure, or an empty string.