Accredited

A LEVEL

Specification

COMPUTER SCIENCE

H446
For first assessment in 2017

ocr.org.uk/alevelcomputerscience

OCR
Oxford Cambridge and RSA

Variables

Variables are assigned using the = operator.

```
x=3
name=“Bob”
```

A variable is declared the first time a value is assigned. It assumes the data type of the value it is given.

Variables declared inside a function or procedure are local to that subroutine.

Variables in the main program can be made global with the keyword `global`.

```
global userid = 123
```

Casting

Variables can be typecast using the `int str` and `float` functions.

`str(3)` returns `“3”`
`int (“3”)` returns 3
`float (“3.14”)` returns `3.14`

Outputting to Screen

`print(`*string*`)`

Example

```
print(“hello”)
```

Taking Input from User

variable`=input(`*prompt to user*`)`

Example

```
name=input(“Please enter your name”)
```

Iteration – Count Controlled

```
for i=0 to 7
    print(“Hello”)
next i
```

Will print hello 8 times (0–7 inclusive).

Iteration – Condition Controlled

```
while answer!=”computer”
    answer=input(“What is the password?”)
endwhile

do
    answer=input(“What is the password?”)
until answer==”computer”
```

Logical Operators

```
AND OR NOT
```

```
e.g.
while x<=5 AND flag==false
```

Comparison Operators

`==`	Equal to
`!=`	Not equal to
`<`	Less than
`<=`	Less than or equal to
`>`	Greater than
`>=`	Greater than or equal to

Arithmetic Operators

`+`	Addition e.g. `x=6+5` gives 11
`-`	Subtraction e.g. `x=6-5` gives 1
`*`	Multiplication e.g. `x=12*2` gives 24
`/`	Division e.g. `x=12/2` gives 6
`MOD`	Modulus e.g. `12MOD5` gives 2
`DIV`	Quotient e.g. `17DIV5` gives 3
`^`	Exponentiation e.g. `3^4` gives 81

Selection

Selection will be carried out with if/else and switch/case

if/else

```
if entry=="a" then
    print("You selected A")
elseif entry=="b" then
    print("You selected B")
else
    print("Unrecognised selection")
endif
```

switch/case

```
switch entry:
     case "A":
        print("You selected A")
     case "B":1
        print("You selected B")
     default:
        print("Unrecognised selection")

endswitch
```

String Handling

To get the length of a string:

```
stringname.length
```

To get a substring:

```
stringname.subString(startingPosition, numberOfCharacters)
```

NB The string will start with the 0th character.

Example

```
someText=”Computer Science”

print(someText.length)
print(someText.substring(3,3))
```

Will display

```
16
put
```

Subroutines

```
function triple(number)
    return number*3
endfunction
```

Called from main program

```
y=triple(7)
```

```
procedure greeting(name)
    print(“hello”+name)
endprocedure
```

Called from main program

```
greeting(“Hamish”)
```

Unless stated values passed to subroutines can be assumed to be passed by value.
If this is relevant to the question byVal and byRef will be used. In the case below x is passed by value and y is passed by reference.

```
procedure foobar(x:byVal, y:byRef)
    …
    …
endprocedure
```

Arrays

Arrays will be 0 based and declared with the keyword *array*.

```
array names[5]
names[0]="Ahmad"
names[1]="Ben"
names[2]="Catherine"
names[3]="Dana"
names[4]="Elijah"

print(names[3])
```

Example of 2D array:

```
Array board[8,8]
board[0,0]="rook"
```

Reading to and Writing from Files

To open a file to read from openRead is used and readLine to return a line of text from the file.

The following program makes x the first line of sample.txt

```
myFile = openRead("sample.txt")
x = myFile.readLine()
myFile.close()
```

`endOfFile()` is used to determine the end of the file. The following program will print out the contents of sample.txt

```
myFile = openRead("sample.txt")
while NOT myFile.endOfFile()
    print(myFile.readLine())
endwhile
myFile.close()
```

To open a file to write to openWrite is used and writeLine to add a line of text to the file. In the program below hello world is made the contents of sample.txt (any previous contents are overwritten).

```
myFile = openWrite("sample.txt")
myFile.writeLine("Hello World")
myFile.close()
```

Comments

Comments are denoted by //

```
print("Hello World") //This is a comment
```

Object-Oriented

Object oriented code will match the pseudocode listed above with the following extensions:

Methods and Attributes:

Methods and attributes can be assumed to be public unless otherwise stated. Where the access level is relevant to the question it will always be explicit in the code denoted by the keywords.

`public` **and** `private.`

```
private attempts = 3
```

```
public procedure setAttempts(number)
    attempts=number
endprocedure
```

```
private function getAttempts()
    return attempts
endfunction
```

Methods will always be instance methods, learners aren't expected to be aware of static methods. They will be called using object.method so

```
player.setAttempts(5)
```

```
print(player.getAttempts())
```

Constructors and Inheritance
Inheritance is denoted by the `inherits` keyword, superclass methods will be called with the keyword super. i.e. `super.methodName(`*parameters*`)` in the case of the constructor this would be `super.new()`
Constructors will be procedures with the name new.

```
class Pet

    private name
     public procedure new(givenName)
         name=givenName

     endprocedure

endclass



class Dog inherits Pet

    private breed

     public procedure new(givenName, givenBreed)
         super.new(givenName)
         breed=givenBreed
     endprocedure

endclass
```

Constructors and Inheritance

Constructors will be procedures with the name new.

```
class Pet

    private name
    public procedure new(givenName)
        name=givenName

    endprocedure

endclass
```

Inheritance is denoted by the `inherits` keyword, superclass methods will be called with the keyword super. i.e. `super.methodName(`*parameters*`)` in the case of the constructor this would be `super.new()`

```
class Dog inherits Pet

    private breed

    public procedure new(givenName, givenBreed)
       super.new(givenName)
       breed=givenBreed
    endprocedure

endclass
```

To create an instance of an object the following format is used

```
objectName = new className(parameters)
```

e.g.

```
myDog = new Dog(“Fido”, “Scottish Terrier”)
```

HTML

Learners are expected to have an awareness of the following tags. Any other tags used will be introduced in the question.

<html>
<link> to link to a CSS file
<head>

<title>

<body>

<h1> <h2> <h3>

<img> including the src, alt, height and width attributes.

<a> including the href attribute.

<div>