Zambia Centre for Accountancy Studies

NCC Diploma in Computing – Level 4

Test 1 Solutions

1.

Variables names that are allowed

Variable	**Allowed**	**Comment**	**Marks**
theValue	Yes		1
AREA	Yes		1
SizeType	Yes		1
testcases	Yes		1
Integer	Yes		1
getName	Yes		1
double	No	'double' is reserved word in Java	0
Go there	No	White space NOT allowed in variable name	0
setName	Yes		1
2013	No	Variable CANNOT start with numeric values	0
Total			7

Variables names with correct style

Variable	**Style**	**Comment**	**Marks**
theValue	Yes		1
AREA	Yes	Valid if variable is a constant	1
SizeType	No	Variable MUST not start with uppercase letter	0
testcases	No	Subsequent words after first character MUST be upper case	0
Integer	No	Initial word MUST be lower case	0
getName	Yes	Valid	1
double	No	'double' is reserved word in Java	0
Go there	No	White space NOT allowed in variable name	0
setName	Yes	Valid	1
2013	No	Variable CANNOT start with numeric value	0
Total			4

2.

	Marks
a) In more precise terms, Java is a statically-typed language, implying that all variables MUST first be declared before they can be used.	2
b) Java permits the use of code blocks, delimited by curly braces, and additionally blocks nested within other existing blocks.	2
c) Java is a programming language that makes it possible for the problem space to be modelled in the form of objects, with their associated properties – data and behaviours – methods.	2
d) Primitive data types, in Java, are the basic types of data.	2
e) byte, short, int, long, boolean, double, float, char	2
Total	10

3.

	Marks
a) 2	2
b) 2000	2
c) 2012	2
d) 2013	2
e) 13	2
Total	10

4.

	Marks
a) String stringVariable = “I am a String”;	2
b) int theInteger = 2013;	2
c) double theDouble = 400.50;	2
d) boolean theBoolean = true;	2
e) char theCharacter = ‘a’;	2
Total	10

5. Remember operator precedence people –tricky question this was :p

	Marks
a) This is a test	2
b) 8	2
c) true	2
d) Munthu was born 43 items	2
e) 0.5	2
Total	10

6. Marks awarded for explanation and for valid code.

	Marks
a) An array, in Java, is a fixed-sized container object that stores values of a single type.	2
b) int[] intArray = new int[5];	2
c) intArray[0] = 5;	2
d) int intArraySize = intArray.length;	2
e) anyArrayName[anyArrayName.length - 1];	2
Total	10

7. Marks awarded for the definitions and examples for each.

	Marks
a) An exception is an/any event that disrupts the normal execution/flow of a program.	2
b) ■ 'try' blocks encloses code that can potentially throw an exception. ■ 'catch' blocks handle exceptions of the type explicitly indicated in the catch argument. ■ 'finally' blocks always execute statements enclosed within it irrespective of whether or not an exception occurs.	4
c) Sorry! An error occurred. Oho!	4
Total	10

8. Marks awarded for the two definitions and the link between the two.

	Marks
a) sacz	4
b) String	2
c) One	2
d) It is invoked int a static method –the 'main' method. In Java, you cannot invoke a none static method from within a static method.	2
Total	10

9. NOTE: This is NOT the precise method for calculating PAYE tax. The computation is slightly more complex than this.

a) Code for method to compute deductions

```
public static double payeDeductions(double grossEarnings) {

        final double NAPSA = 0.15; // 15%
        double deductions = 0;
        double payeTax;

        if(grossEarnings <= 2000000) {
                payeTax = 0;
        }
        else if (grossEarnings >= 2000001 && grossEarnings <= 2800000) {
                payeTax = 0.25;
        }
        else if (grossEarnings >= 2800001 && grossEarnings <= 5700000) {
                payeTax = 0.30;
        }
        else {
                payeTax = 0.35;
        }

        deductions = (NAPSA*grossEarnings) + ((grossEarnings -
                (NAPSA*grossEarnings))*payeTax);

        return deductions;
}
```

b) Code for method to compute NET pay

```
public static double payeNETPay(double grossEarnings) {

        return grossEarnings - payeDeductions(grossEarnings);
}
```

10.

a) Code for the two constructors

```
public Student() {
        studentName = "Munthu";
        studentNumber = 1111111111;
}

public Student(String studentName, int studentNumber) {
        this.studentName = studentName;
        this.studentNumber = studentNumber;
}
```

b) Code for getter methods

```
public String getStudentName() {
        return studentName;
}

public int getStudentNumber() {
        return studentNumber;
}
```

c) Code for object

```
        Student johnBanda = new Student("John Banda", 2222222222);
```

d) Code for accessing student name

```
        System.out.println(johnBanda.getStudentName());
```