| Please fill in your Student Number and, optionally, Name. | For Official Use |
|---|---|
| **Student Number** : _____ | **Mark** : _____ |
| **Name** : _____ | **Marker** : _____ |

**University of Cape Town ~ Department of Computer Science**

**Computer Science 1015F ~ 2007**

# Theory Test 2A

**Marks** : **30**

**Time** : **40 minutes**

**Instructions:**

a) Answer all questions.

b) Write your answers in the space provided.

c) Show all calculations where applicable.

**Question 1: Multiple Choice. [5]**

**For each question, write down ONLY the letter of the correct answer.**

a) The while stament is [1]

A. A branching mechanism

B. A loop mechanism

C. A Boolean expression

D. All of the above

Answer: _____

*A*

b) Examine the following Java code: [1]

```
String a = "right",b= "left" ;
b = (a.length() <10) ? a : "overrun";
System.out.println(b);
```

After executing this code, the value of the variable **`b`** is:

A. “right”

B. “left”

C. “overrun”

D. “a.length()”

Answer: _____

*A*

c) Which of the following operators has the *highest precendence*? [1]

A. &

B. ||

C. &&

D. =

Answer: _____

*A*

d) Examine the following Java code: [1]

```
boolean A=true, B=false, C=true, D=false;
System.out.print(A || B && C || D);
System.out.print( ! D && C);
```

When executing this code, the output is:

A. truetrue

B. truefalse

C. falsetrue

D. falsefalse

Answer: _____

*A*

e) Examine the following Java code: [1]

```
int i=4;
System.out.print(i++ + " ");
System.out.println(++i);
```

When executing this code, the output is:

A. 5 5

B. 5 6

C. 4 4

D. 4 6

Answer: _____

*D*

**Question 2: Short questions [6]**

a) Explain why the following Java code does NOT result in a run-time error.

```
int apples =0;
double horses =15.0;
if((apples>0)&&(horses/apples>1))
    System.out.println("Every horse has an apple!");
```

[2]

*The && expression in Java uses lazy evaulation/ short-cut evaluation. In an "AND" expression, if the first operand evaluateds to false, the second operand is not evaluated. In this case, apples>0 evalautes to false, so horses/apples is never calculated and no run-time error for dividing by zero occurs.*

b) What is an *inifite loop*?

[1]

*A loop that does not stop – it runs forever.*

c) Write down a n example of Java code that will result in an *inifite loop*.

[1]

*Any correct example, e.g*

```
for(int i=10;i>0;i++)
System.out.println(i);
```

a) Describe how you would rewrite the following code to use a `while` statement instead of a `do-while` statements. Note that the new code must behave exactly as the old version.

```
double x;
 do {
   x = Math.random();
   if(x<0.5)
     System.out.println("Heads");
   else
     System.out.println("Tails");
 } while (x<0.5);
```

[2]

*The while loop will look the same, except that the loop body will have to be repeated above the while statement, to ensure that it is executed at least once.*

*Question 3:Longer questions[9]*

b) Rewrite the following `switch` statement as nested `if-else` statements.

```
switch(choice)
  {
  case 1: System.out.println("A");
        break;
  case 2:
  case 3: System.out.println("B");
         break;
  case 4: System.out.println("C");
         break;
  default: System.out.println("Z");
  }
```

[3]

```
 if(choice==1)
        System.out.println("A");
      else if((choice==2)||(choice==3))
        System.out.println("B");
      else if(choice==4)
        System.out.println("C");
      else
        System.out.println("D");
```

a) Now write a program to draw an arrow of a certain height, supplied by the user.

e.g. If the user supplied a height of 1, the output will be:

```
*
```

If the user supplied a height of 2, the output will be:

```
***
 *
```

If the user supplied a height of 3, the output will be:

```
*****
 ***
  *
```

And so on. You are given the outline of the program, just supply the missing lines of code.

```
import java.util.Scanner;
  public class mystery
  {
    public static void main(String[] args)
    {
     Scanner keyboard = new Scanner(System.in);
     System.out.println("Enter the height of the triangle:");

          int height = keyboard.nextInt();
          for(int row=1; row<=height; row++)
          {
           for(int spaces=0; spaces<row; spaces++)
             System.out.print(' ');

           for(int stars=(height-row)*2-1; stars>0; stars--)
             System.out.print('*');

           System.out.println();
           }


    }
  }
```

[6]

**Question 4: Testing [5]**

a) Explain the difference between path coverage and statement coverage? [2]

*Path coverage involves testing every possible combination of statements that is executed in order [1]. Statement coverage involves testing only that every statement is executed once [1].*

b) Suppose you are testing the following program. Based on equivalence classes and boundary values, provide a set of 9 test values that may be used. [3]

```
if (x<100)
  // do something
else if (x<200)
  // do something else
else
// do something completely different
```

*50, 150, 250 [1]*

*99, 100, 101 [1]*

*199, 200, 201 [1]*

## Question 5: Object Oriented Programming [5]

a) Why do we use object oriented programming? [1]

*OOP code organisation promotes better understanding and maintainability. [1]*

b) What is the difference between a class and an instance? [2]

*Class is a template to create objects [1] – instance is an object created according to the definition provided by this template. [1]*

c) What is a method? How is a method invoked on a method? [2]

*Method is a named set of statements that can be invoked on an object/instance. [1]*

*Use dot notation i.e., object.method (arguments); [1]*