###### RMIT – Programming 1

**Mid Semester Test (2008 Sem1; Monday, 1 May 2008)**

**Sample Solutions & Marking Guide**

**Part A Complete the following code fragments (55 marks)**

1. **(a) (10 marks) ; (b) (5 marks)**

int term\_i = 1; System.out.println(term\_i);

**int sum = term\_i; // Part (b) solution**

for (int i = 2; i <= **20**; i++)

{

**term\_i = term\_i \* 3;**

**sum += term\_i;**

**System.out.print(“ “ + term\_i); // Part (a) solution** }

**System.out.println(sum);**

**MARKING GUIDE**

1. **3 marks for correct loop test expression; 2 marks for print (print or println will do); 3 marks for expression to print the squares.**
2. **2 marks for initialisation/print; 3 marks for loop body update**
3. **(10 marks)**

**&\***

**?&&\*\***

**??&&&\*\*\***

**???&&&&\*\*\*\***

**????&&&&&\*\*\*\*\***

**MARKING GUIDE**

**2 marks for getting each line right. Deduct 0.5 for each error but no more than 1.5 per line if they have recognised that spaces (?), &s and \*s should appear on each line as appropriate.3. (15 marks). MARKING GUIDE is embedded in sample solution below.**

import java.util.\*;

public class Palindrome

{

public static void main(String args[])

{

int n = 0; // Number of strings

String s[] = new String[50]; // space for 50 strings

String rev;

Scanner sc = new Scanner(System.in);

for(int i=0; i < **50**; i++){ **// 1 mark**

System.out.print("Enter string " + (i+1)+" : ");

s[i] = sc.nextLine();

**// Write code to terminate loop if empty**

**// line has been entered**

**if (s[i].length() == 0) // 3 marks**

**break;**

}

n = i – 1; // n assigned to number of strings entered

**// Complete loop to process n strings in array s**

for (int i = 0; i < **n**; i++) { **// 2 marks**

**// Convert ith string s[i]to lower case**

**s[i] = s[i].toLowerCase(); // 3 marks**

**// Store the reverse of s[i] in string rev**

**String temps = ““; // 3 marks (including loop)**

**for (int j = s[i].length() – 1; j >= 0; j--)**

**temps += s[i].charAt(j);**

**// Compare s[i] and rev. If s[i] and rev are**

**// equal report s[i]as being a palindrome,**

**// otherwise report it as not being a palindrome**

**// 3 marks**

**System.out.println(“The string: “ + s + “ is “);**

**if (!temps.equals(s))**

**System.out.print(“not “);**

**System.out.println(“a palindrome. “);**

}

}

}

**4.**

public class Exceed50andAverage {

public static void main (String[] args) {

int M[] = {23,55,50,67,34,98,50};

double sum = 0.0, average;

**// Write the code here**

System.out.println(“Elements > 50:”);

for (int i = 0; **i < M.length; i++)**

{

**sum += M[i];**

**if (M[i] > 50)**

**{**

**System.out.print(i+":"+M[i]+" ");**

**}**

}

**average = sum/M.length;**

System.out.println(“Average mark= “ + average);

}

}

# MARKING GUIDE

# 4 marks for completing loop expressions;

# 2 marks for updating sum inside loop;

# 6 marks if-statement and print;

**3 marks for computation of average, outside loop;**

# Part B (45 marks) A Program to keep track of Animals

Section I – Defining the Animal class

public class Animal {

public Animal(String n, double w, Animal mum)

{

name = n;

weight = w;

mother = mum;

}

**// Add new constructor here (Part (ii))**

**public Animal(String id, String n, double w, Animal mum)**

**{**

**Animal(n, w, mum);**

**ID = id;**

**}**

public void changeWeight(double w) { weight += w;}

public void setMother(Animal m) { mother = m;}

public double getWeight() { return weight;}

public String getName() { return name;}

public Animal getMother(){return mother;}

**// Insert new methods here (Parts (iii)—(v))**

**public void setWeight(double w) { weight = w; }**

**public String getID() { return ID; }**

**public void print() {**

**System.out.println(“ID: “ + ID);**

**System.out.println(“Name: “ + name);**

**System.out.println(“Weight: “ + weight);**

**System.out.print(“Mother ID/Name: “);**

**if (mother == null)**

**System.out.println(“Mother unknown “);**

**else**

**System.out.println(mother.getID() + “/“ +**

**Mother.getName());**

**}**

private String name;

private double weight;

private Animal mother;

**// Add new instance variable here (Part (i))**

private String ID;

}

Section II – Using the Animal class

public class ManageZoo

{

public static void main(String args[]){

**// (i) 3 marks (1.5 for LHS, 1.5 for RHS)**

**Animal e[] = new Animal[3];**

**// (ii) 6 marks (2 each: 0.5 LHS; 0.5 new; 1.0 args)**

**e[0] = new Animal(“A123“, “OldTiger“, 34, 100, null);**

**e[1] = new Animal(“A124“, “MummyChimp“, 20, 50, null);**

**e[2] = new Animal(“A125“, “BabyChimp“, 2, 4, e[1]);**

**// (iii) 16 marks;**

**Scanner console = new Scanner(System.in); // 2 marks**

**System.out.println("Enter ID Op and amount"); // 2 marks for**

**// read/exit loop**

**while ( true )**

**{**

**int i;**

**String ID; // 3 marks for extracting ID/op/amt**

**String s = console.next();**

**if ( s.compareTo("X") == 0 )**

**{**

**System.out.println("Finished processing transactions");**

**break;**

**}**

**String op = console.next();**

**double wt = Double.parseDouble(console.next());**

**Animal an = null;**

**for (i=0; i<3; i++) // 3 marks for extracting an**

**if ( e[i].getID().compareTo(ID) == 0)**

**{**

**an = e[i];**

**break;**

**}**

**if ( an == null) // 2 marks for error case**

**System.out.println("No animal found with ID" + ID);**

**else**

**if ( op.compareTo("S") == 0) // 2 marks for setting salary**

**{**

**an.setWeight(wt);**

**System.out.println("New weight for animal with ID" + ID);**

**}**

**else**

**if ( op.compareTo("U") == 0) // 2 marks for updating salary**

**{**

**an.raieWeight(amt);**

**System.out.println("Update weight for animal with ID " + ID);**

**}**

}

**// (iv) 5 marks; 2 for loop; 3 for loop body**

**for (int j=0; j < 3; j++)**

**e[j].print();**

}

2