

CSE1007 – JAVA PROGRAMMING

ASSESSMENT NO.02

DEADLINE: -29/09/2022

1. Write the following two methods

// Return the reversal of an integer, i.e. reverse(456) returns 654

public static int reverse(**int** number)

// Return true if number is palindrome

public static boolean isPalindrome(**int** number)

Use the **reverse** method to implement **isPalindrome**. A number is a palindrome if its reversal is the same as itself. Write a test program that prompts the user to enter an integer and reports whether the integer is a palindrome.

2. Write a method that finds the smallest element in an array of integers using the following header:

public static double min(**double**[] array)

Write a test program that prompts the user to enter ten numbers, invokes this method to return the minimum value, and displays the minimum value.

3. Design a class named **Stock** that contains:

- A string data field named symbol for the stock's symbol.
- A string data field named name for the stock's name.
- A double data field named previousClosingPrice that stores the stock price for the previous day.
- A double data field named currentPrice that stores the stock price for the current time.
- A constructor that creates a stock with specified symbol and name.
- A method named getChangePercent() that returns the percentage changed from previousClosingPrice to currentPrice.

Implement the class. Write a test program that creates a Stock object with the stock symbol JAVA, the name Sun Microsystems Inc, and the previous closing price of 4.5. Set a new current price to 4.35 and display the price-change percentage.

Stock	
symbol: String	The symbol of this stock.
name: String	The name of this stock.
previousClosingPrice: double	The previous closing price of this stock.
currentPrice: double	The current price of this stock.
Stock(symbol: String, name: String)	Constructs a stock with a specified symbol and a name.
getChangePercent(): double	Returns the percentage of change of this stock.

4. Design a class named **Account** that contains:

- A private **int** data field named **id** for the account (default **0**).
- A private **double** data field named **balance** for the account (default **0**).
- A private **double** data field named **annualInterestRate** that stores the current interest rate (default **0**). Assume all accounts have the same interest rate.
- A private **Date** data field named **dateCreated** that stores the date when the account was created.
- A no-arg constructor that creates a default account.
- A constructor that creates an account with the specified id and initial balance.
- The accessor and mutator methods for **id**, **balance**, and **annualInterestRate**.
- The accessor method for **dateCreated**.
- A method named **getMonthlyInterestRate()** that returns the monthly interest rate.
- A method named **withdraw** that withdraws a specified amount from the account.
- A method named **deposit** that deposits a specified amount to the account.

Implement the class. Write a test program that creates an **Account** object with an account ID of 1122, a balance of \$20,000, and an annual interest rate of 4.5%. Use the **withdraw** method to withdraw \$2,500, use the **deposit** method to deposit \$3,000, and print the balance, the monthly interest, and the date when this account was created.

5. Some Websites impose certain rules for passwords. Write a method that checks whether a string is a valid password. Suppose the password rule is as follows:

- A password must have at least eight characters.
- A password consists of only letters and digits.
- A password must contain at least two digits.

6. Write a program that prompts the user to enter a password and displays "**Valid Password**" if the rule is followed or "**Invalid Password**" otherwise.

The String class is provided in the Java library. Provide your own implementation for the following methods (name the new class MyString1):

```
public MyString1(char[] chars);
```

```
public char charAt(int index);
```

```
public int length();
```

```
public MyString1 substring(int begin, int end);
```

```
public MyString1 toLowerCase();
```

```
public boolean equals(MyString1 s);
```

```
public static MyString1 valueOf(int i);
```