Java Cheat Sheet – DoubleDAmarasinghe @ github

Primitive type

```
INTEGER: byte(8bit), short(16bit), int(32bit), long(64bit) DECIMAL:
float(32bit), double(64bit) OTHER: boolean(1bit), char(unicode)
```

Primitive Operators

```
Assignment operator: “=”
Binary operator: “+ - * / %”
Unary operator: “+ - ++ --”
Boolean not operator: “!”
Boolean binary: “== != > >= < <=”
Boolean binary only: && ||
Bitwise operator:~ & ^ | << >> >>>
Ternary operators: bool true false
```

None-Primitive type

```
String, Array, Class, Interface
```

Widening casting

Smaller to larger type

```
int myint = 9;
double mydouble = myint;
System.out.println(mydouble) //9.0
System.out.println(myint) //9
```

Narrowing casting

Larger to smaller

```
double mydouble = 9.7;
int myint = (int) mydouble;
System.out.println(mydouble) //9.7
System.out.println(myint) //9
```

Conversions

```
int x = Integer.parseInt(“999”);
float y = Float.parseFloat(“9.9”);
String abc = Integer.toString(12);
String xyz = Float.toString(0.54);
```

import java.util.Scanner;

```
Scanner s =new Scanner(System.in);
String username = s.nextLine();
System.out.println(username);

nextBoolean(), nextByte(), nextDouble(), nextFloat(), nextInt(),
nextLong(), nextShort()
```

if, else if, else statement

```
if(condition){statements}
else if(condition){statement}
else{statement}
```

For loop

```
for(int i=0; i<10; i++)
{
    System.out.println(i)
}
```

While loop

```
int i=0;
while(i<10)
{
    System.out.println(i);
    i++;
}
```

Do while loop

```
int i=0;
do
{
    System.out.println(i);
    I++;
}
while(i<10);
```

Switch statement

```
int x = 3;
switch (x)
{
    case 1:
     System.out.println(“A”);
     breake;

    case 2:
     System.out.println(“B”);
     breake;

    case 3:
     System.out.println(“C”);
     breake;

    default:
     System.out.println(“D”)
}
```

Arrays

```
int [] x = new int [10]; //ten 0s
int [][] x = new int [5][5];
//5 by 5 matrix
int [] x = {1,2,3,4,5};
x.length; // 5
int [][] x = {1,2},{3,4,5,};
//ragged array
String [] y = new String [10]; //ten nulls
```

Functions/Methods

Static declarations

```
public static int myFun(){smt}
private static double myFun(){smt}
static void myFun(){smt}
```

Instance declarations

```
public void myFun(){smt}
private int myFun(){smt}
```

Call a Function/Method

```
public static void main(String[] args)
{
    myFun();
}
```

Class

```
Class Myclass;
```

Object

```
public class Myclass
{
    Int x = 5;

    Public static void main(String[] args)
   {
     Myclass myobj = new Myclass()
     System.out.println(myobj.x);
   }
}
```

Imports in GUI

```
java.awt.Color;
java.awt.Font;
java.awt.event.ActionEvent;
java.awt.event.ActionListener;
java.awt.event.KeyEvent;
java.awt.event.KeyListener;
java.sql.*;

javax.swing.BorderFactory;
javax.swing.border.Border;
javax.swing.*;
javax.swing.table.DefaultTableModel;
```

Java inheritance

```
public class Window extends JFrame{}
```

Creating objects

```
JFrame f1 = new JFrame();
JPanel p1 = new JPanel();
JButton b1 = new JButton(“Exit”);
JLabel l1 = new JLabel(“Name”);
JTextArea t1 = new JTextArea();
JTextField tf1 = new JTextField();
JRadioButton r1 = new JRadioButton(“Male”);
JComboBox c1 = new ComboBox(“Maried”);

DefaultTableModel model = new DefaultTableModel();
JTable jt = new JTable(model);

Font f1= new Font("Arial", Font.BOLD, 36);

Border b1 = BorderFactory.createLineBorder(Color.black, 2);

Color clr = new Color(0,255,0);

JMenuBar jmb = new JMenuBar();
JMenu jm = new JMenu("File");
JMenuItem jmi1 = new JMenuItem("Exit");

JOptionPane jp = new JOptionPane();
```

Call a Function/Method

```
public ItemSeller(){}
```

Frame attributes

```
frame1.setSize(1360,720);
frame1.setTitle("Item_Seller");
frame1.setLayout(null);
frame1.setLocationRelativeTo(null);
frame1.setVisible(true);
frame1.setDefaultCloseOperation(EXIT_ON_CLOSE);
frame1.setResizable(false);
```

Panel attributes

```
mainpanel.setSize(1360,720);
mainpanel.setLayout(null);
mainpanel.setVisible(true);
mainpanel.setBackground(new Color(25,130,146));
```

Label attributes

```
ammount.setLocation(365,425);
ammount.setSize(600,100);
ammount.setFont(amunt);
ammount.setForeground(Color.white);
```

Text area attributes

```
jtxitem.setSize(200,30);
jtxitem.setLocation(120,135);
jtxitem.setBorder(brd);
jtxitem.setFont(itempanelfont);
```

Set JTable null

```
model.setRowCount(0);
```

Table attributes

```
jt.setLocation(370,110);
jt.setSize(720,300);
jt.setBorder(brdjt);
jt.setModel(model);
jt.setFont(itempanelfont);
```

Model attributes

```
model.addColumn("A");
model.addColumn("B");
model.addColumn("C");
model.addColumn("D");
```

Button attributes

```
delete.setSize(200,30);
delete.setLocation(25,135);
delete.setBorder(brd);
delete.setFont(itempanelfont);
delete.addActionListener(new ActionListener()
    {
       public void actionPerformed(ActionEvent arg0)
        { jtx.setText(null);
          jtxitem.requestFocus();
        }
    });
```

Set shortcut key for button

```
reset.setMnemonic('z');
```

Text field attributes

```
jtxitem.setSize(200,30);
jtxitem.setLocation(120,135);
jtxitem.setBorder(brd);
jtxitem.setFont(itempanelfont);
jtxitem.addKeyListener(new KeyListener()
{
    @Override
    public void keyPressed(KeyEvent evt)
      {
          int a = evt.getKeyCode();
          if (a==10) //enter key
           {
            jtxqty.requestFocus();
           }
          int b = evt.getKeyCode();
          if (b==KeyEvent.VK_SPACE) //space key
           {
            jtxpay.requestFocus();
           }
      }

    @Override
    public void keyReleased(KeyEvent arg0) {}

    @Override
    public void keyTyped(KeyEvent arg0) {}
});
```

Import data from sql database

```
jtxitem.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
      {
        try
         {
          Class.forName("com.mysql.jdbc.Driver");
          Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost/doubled",
"rona","rona");
          Statement s=conn.createStatement();
          String sql = "select*from user where
code="+jtxitem.getText()+"";
          PreparedStatement ps = conn.prepareStatement(sql);
          ResultSet rs = ps.executeQuery();
          while(rs.next())
               {
//shows item name in item namelabel
itemname.setText(rs.getString(1));
                  //shows item price in itemprice label
itemprice.setText("Rs."+rs.getString(5));
               }
       }
         catch(Exception e1)
           {
            System.out.println(e1.getMessage());
           }
       }
 } );
```

Exit button

```
jmi1.addActionListener(new ActionListener()
{
  public void actionPerformed(ActionEvent arg0)
     {
         System.exit(0);
     }
});
```

Export data to sql database

```
jbtadd.addActionListener(new ActionListener()
{
  public void actionPerformed(ActionEvent e)
   {
     try
       {
          Class.forName("com.mysql.jdbc.Driver");
          Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost/","rona","rona");
          Statement s=conn.createStatement();
          s.execute("create database doubled;");
          s.execute("use doubled;");
          s.execute("create table user(name varchar(25), code
varchar(10), q varchar(100), cost decimal(10), price decimal(10),
edate varchar(100), pdate varchar(100), sup varchar(100), re
varchar(100));");
          PreparedStatement ps = conn.prepareStatement("insert into
user(name,code,q
,cost,price,edate,pdate,sup,re)values(?,?,?,?,?,?,?,?,?);");
          ps.setString(1,jtx1.getText());
          ps.setString(2,jtx2.getText());
          ps.setString(3,jtx3.getText());
          ps.setString(4,jtx4.getText());
          ps.setString(5,jtx5.getText());
          ps.setString(6,jtx6.getText());
          ps.setString(7,jtx7.getText());
          ps.setString(8,jtx8.getText());
          ps.setString(9,jtx9.getText());
          int x = ps.executeUpdate();
          if(x>0)
            {
              System.out.println("Work Work");
            }
           else
            {
              System.out.println("Nooo");
            }
          }
        catch(Exception e1)
          {
              System.out.println(e1.getMessage());
          }
} );
```

Add components

```
itempanel.add(itemprice);
mainpanel.add(jt);
itempanel2.add(reset);
logopanel.add(companyname);
frame1.add(mainpanel);
frame1.setJMenuBar(jmb);
jmb.add(jm);
jm.add(jmi1);
```

Main method and creating object

```
public static void main(String[]args)
  {
    ItemSeller obj = new ItemSeller();
  }
```

Optionpane massage

```
jp.showMessageDialog(frame1,"Item
Sold!","Alert",JOptionPane.INFORMATION_MESSAGE);
```

Add sql data into JTable

```
jtxqty.addActionListener(new ActionListener() {
  public void actionPerformed(ActionEvent e)
    {try
      {
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost/doubled","rona",
"rona");
        Statement s=conn.createStatement();
        String sql = "select*from user where
code="+jtxitem.getText()+"";
        PreparedStatement ps = conn.prepareStatement(sql);
        ResultSet rs = ps.executeQuery();
while(rs.next())
{
          //get data from database
          String a = rs.getString(1);
          String b = rs.getString(2);
          Float itemprice1 = Float.parseFloat(rs.getString(5));
          //get contains of jtxqty and convert into float
          Float itemqtys = Float.parseFloat(jtxqty.getText());
          String c = Float.toString(itemprice1*itemqtys);
          String d = jtxqty.getText();
          //adding data row from database to jtable
          model.addRow(new Object[] {b,a,c,d});
}
}    catch(Exception e1)
{       System.out.println(e1.getMessage());
}} });
```