**Goal: This program sets up a nail appointment printing a receipt at the end displaying appointment details including the total. It will prompt the user to select the service options they want and an add on if they would like. Then it will prompt the user to enter the name for the appointment and to select a date and time from the available options. Once the user successfully selects an available option the program will print out a receipt displaying the appointment details and total cost of the services they selected, the receipt will also print to a text file.**

**Service.java**

**BEGIN** interface Service {

Public double to get cost for a service()

Public void setCost (to set cost to double cost variable)

Public String to get Name of service()

Public void setName (to set name of service to name variable)

} **END** of interface Service

**Order.java**

**BEGIN** interface Order{

public String to get Date()

public void setDate (to date selected by customer(String date)

Public String to get Customers name()

public void set customers name (to string variable customerName)

public double get total cost()

public void set total cost(to double total cost variable)

} **END** of interface Order

**Salon.java**

**BEGIN** class SalonOrder (implementing interface Order) {

**Create** Services array from SalonService array

**Set** Service index to 0

**Create** extraServices array from SalonService class

**Set** extraService index to 0

**Declare** variables from interface Order

**Set** DateTimeFormatter to the date format

**Declare** variables for an array to set up available appointment times

Int days, start time, num of hour slots

**Create** a String time array[] = new String [days \* num of hour slots]

**Set** time array index to 0

**Method** to generate dates{

**Create** for loop {

**Set** local date time( variable today) to America/LosAngeles

**Set** local date time (time variable for today) to (today, midnight)

**Set** local date time (time variable for tomorrow) to (time variable for today)

**Create** for loop {

Format the date to = time variable for tomorrow plus hours from dateformat

**Create** time array [index of time] = date format

Increasing index

}

}

**Return** this time array to later ask user for input

}

**Create** method to add a Service to (SalonService service) {

**Set** this Services [service index] = the service

Increasing Service Index++

}

**Create** method to add an extra Service to (SalonService service) {

**Set** this extraServices [service index] = the service

Increasing Service Index++

}

// **Add** methods and return the services for methods in the interface Service

**Method** to get the month {

Return the month they entered

}

**Method** to set month {

The month they entered = month variable

}

**Method** to get the day {

Return the day they entered

}

**Method** to set day {

The day they entered = day variable

}

**Method** to get the date {

Return the date they selected

}

**Method** to set date {

The date they selected = day variable;

}

**Method** to get the name of customer{

Return the name of customer they entered

}

**Method** to set the name of the customer {

The name of customer entered = customer name variable

}

**Method** to get the the total cost () {

Set total variable to 0

// Loop through all services they entered to get the cost of each

for (int i = 0; i less than or equal to the Service index -1; increasing each loop) {

Add service [i] cost with method get cost to total cost

}

// Loop through all extraservices they entered to get the cost of each

for (int i = 0; i less than or equal to the extraService index -1; i++) {

Add extraservices [i] cost with method get cost to total cost

}

return total

}

**Method** to set total cost {

This total cost added = total cost variable;

}

**Method** to print the services() {

**Create** decimal format for prices to use in the menu

**Print Prompt** “Here is your receipt.”

**Print Prompt “** —-----------------------------------------------”

**Print Prompt** “Zippy's Nail Salon Appointment\r\n”

**Print Prompt** “Appointment for: " + customer name variable”

**Print Prompt** “Appointment Date: " + call get date”

for (loop through the service index until we get to the service they chose) {

**Print Prompt** “ Service: " + call service get name we pulled from loop

}

for (loop through the extraService index until we get to the extraService they chose) {

**Print Prompt** “Add-on selected: : " + call extraService get name we pulled from loop

}

**Print Prompt** “Total: $" + df.format(call get total cost) + "\r\n"

**Print Prompt** “ Thank you for booking with Zippy's.”

**Print Prompt** “See you soon!”

**Print Prompt “** —-----------------------------------------------”

**Print Prompt to text file** with String data = "\r\n------------------------------------------- \r\n";

data += "Zippy's Nail Salon Appointment\r\n";

data += "Appointment for: " + call customers name + "\r\n" +

"Appointment Date: " + call get Date + "\r\n”

for (loop through the service index until we get to the service they chose) {

**Print Prompt** “ Service: " + call service get name we pulled from loop

}

for (loop through the extraService index until we get to the extraService they chose) {

**Print Prompt** “Add-on selected: : " + call extraService get name we pulled from loop

}

**Print Prompt to text file**“Total: $" + df.format(call get total cost) + "\r\n"

**Print Prompt to text file** “ Thank you for booking with Zippy's.”

**Print Prompt to text file** “See you soon!”

**Print Prompt to text file “** —-----------------------------------------------”

**// Print receipt to text file**

writeCustomerReciept writer = new writeCustomerReciept(data, "receipt.txt");

writer.writeToFile(writer.getTextToWrite(), writer.getFileName());

} **END** class SalonOrder

**BEGIN** class SalonService (implementing interface Service){

**Declare** variables from interface Service

Cost of service variable

name of service variable

SalonService(name of service variabel, cost of service variable) {

**Declare** this setCost as cost of service variable

**Declare** this setName as name of service variable;

}

**Method** to get the Cost of service {

Return this cost of service

}

**Method** to set Cost {

This cost = cost of service variable

}

**Method** to get the service name {

Return this service name

}

**Method** to set service name {

This service name = name of service variable

}

}**END** SalonServie

**BEGIN** class receiptPrinter {

writeTo File(String textToWrite, String filename) {

**Set** variable data to = textToWrite

**Set variable** currentDirectory to = user directory

// Print out user Directory to show where the receipt will print

**Print prompt** “ Current dir: “ + currentDirectory variable

file = new File(currentDirectory + "/" + fileName);

fop = new FileOutputStream(file, true);

**Function** If (file does not exist) {

Create a new file

}

// **Get** the content in bytes

byte[] contentInBytes = data.getBytes();

Fop write the contentInBytes()

flush()

close()

} catch (IOException c) {

// print

c.printStackTrace();

// print getMessgae

**Print prompt** (c.getMessage());

} finally {

try {

// if it does not equal null then close

if (fop != null) {

fop.close();

}

} catch (IOException c) {

c.printStackTrace();

}

}

} **END** writeToFall

} **END** class receiptPrinter

**BEGIN** class writeCustomerReciept (extends receiptPrinter class){

**Declare** variables filename, textToWrite;

public writeCustomerReciept(variable textToWrite, variable fileName) {

This textToWrite = the variable textToWrite;

This fileName = the variable fileName;

}

**Method** to getFileName {

return this fileName

}

**Method**  to get TextToWrite {

return this textToWrite

}

} **END**  class writeCustomerReciept

**BEGIN** class Salon {

**Create** decimal format for prices to use in the menu

**Create** new SalonService services array for the 6 service

**Create** new SalonService extraService array for the 2 add-on’s

// **Create** constructor for Salon class that runs on instance creation {

/\* this actual name and price is stored here so when the customer selects this option and it will loop through the array to the certain index and print out the name and price \*/

This.services[0] = a new SalonService (“Actual service name”, price)

// continue until it fills the array of 6

// for extraservice (add-on)

This.extraService[0] = new SalonService (“Actual service name”, price)

**Set** startOrder

}

**Method** showServices prints out options {

For ( the length of the services print the following) {

**Print Prompt** “Option " + i + " " + get name for service selected + " $" + format the service cost

}

**Print Prompt**"\r\nAvailable add-on's:"

for (the length of the extraservices print the following) {

**Print Prompt** "Option " + i + " " + get name for extra service selected + " $" +

format the extraservice cost

}

**Print Prompt** "\r\nPlease select an option: "

} **END** of showService

**Method** showExtraServices adds other service to order {

for(length of extraservice) {

**Print Prompt** "Option " + i + " " + get name for extra service selected + " $" + format the extraservice cost

}

}**END** of showExtraServices

**Method** startOrder {

Salon order = a new salon order

**Print Prompt** "\r\nWelcome to Zippy's Nail Salon! "

**Print Prompt** "We are open everyday from 9:00 AM to 1:00 PM\r\n"

**Create** variable for a new service

**Create** Do {

**Create** Do{

**Print Prompt** "Services we offer: ")

**Call** showServices();

**Construct** Scanner object from System.in for option (integer)

**Validate**

If ( option is smaller than zero or bigger than 5) {

Loop {

**Print Prompt** “\r\n" + option + " is not a valid option. Please select from the following: "

**Call** showServices

**Create** new option for input

} while options is smaller than zero or greater than 5

**Print Prompt** "You selected option: " + option

} **END** of if

Add valid service to the order (order.addService)

**Print Prompt** "\r\nWould you like an add-on? (Yes or No)"

**Create** new string for add on

} **END**  while ( add extra service equals yes or Yes)

If (string is yes or YES) {

**Call** showextraService to print out add-on menu

**Create** new int variable for add on option they selected

**Validate**

if( new in variable equals 0 or equals 1) {

**Create** switch (new int variable)

Case 0:

**Print Prompt** "Add-on selected: Nail design"

break

Case 1:

**Print Prompt** "Add-on selected: Extra 10 minute massage."

}

} else {

do {

**Print Prompt** "\r\n" +new in variable+ " is not a valid option. Please select from the following: ");

**Call** showExtraServices

**Create** new int for their add on option

} loop while (new in variable is less than 0 or greater than 1)

}

**Add** the extra service to the order (order.add)

} else if (new int variable equals No or no) {

**Print Prompt** ("No add on selected.");

} else {

User did not enter yes/Yes or No/no

**Print Prompt** "Invalid answer, please type yes to select an add-on or no for no add-on."

**Create** new string for add on

}

**Reset** sc.reset

**Print Prompt** "\r\nName for appointment: "

**Create** new string for customer name

**Set** new string for customer name to (customername variable)

// generate dates and fills in local string array of dates

String[] dateOptions = order.generateDates();

**Print Prompt** "Select a date option below: "

// prints out options

for (int dateIndex = 0; dateIndex < dateOptions.length; dateIndex++) {

**Print Prompt** System.out.println(dateIndex + ") " + dateOptions[dateIndex]);

}

**Declare** int variable for they date they picked

do {

Int variable for date they picked entered in the next line

}loop while int variable they picked for date is greater than dateOptions length or is less than zero

**Store**  int variable they picked for date in order (order.setDate)

**Print** their order order.printServices()

} **END** of start Order

} **END** class Salon

**BEGIN**  class Main {

Public static void main(String arg[]) {

Salon testService = new Salon();

}

}