```
class Player
  {
     public String name;
     public Weapon[] backpack;
     public int numItems;
     public double money;

     public Player(String n, double m)
     {
        name = n;
        money = m;
        numItems = 0;
        backpack = new Weapon[10];
     }

     public void buy(Weapon w)
     {
        System.out.println(w.weaponName+" bought...");
        backpack[numItems] = w;
        numItems++;
        System.out.println(numItems);
     }
     public void withdraw(double amt)
     {
        money = money - amt;
     }

     public boolean inventoryFull()
     {
        return (numItems == 10) ;
     }

     public void printCharacter()
     {
        System.out.println(" Name:"+name+"\n Money:"+money);
        printBackpack();
     }

     public void printBackpack()
     {
         System.out.println(" "+name+", you own "+numItems+" Weapons:");
        for (int x = 0; x < numItems; x++)
        {
            System.out.println(" "+backpack[x].weaponName);
        }
        System.out.println();
     }
  }
```

```
class Weapon
   {
      public String weaponName;
      public int range;
      public int damage;
      public double weight;
      public double cost;

      public Weapon(String n, int rang, int dam, double w, double c)
      {
         weaponName = n;
         damage = dam;
         range = rang;
         weight = w;
         cost = c;
      }
   }


public class ShopItem {
   Weapon item;
   int numberInStock;

   public ShopItem(Weapon w, int nInStock){
      item=w;
      numberInStock=nInStock;
   }
}
```

```
public class ArrayManager {

	int maxItems;	// records the max size of the table
	int numItems;	// records number of items in the list
	ShopItem[] table; //hashtable itself

	public ArrayManager(int size)
	{
	   maxItems = size;
	   numItems = 0;
	   table = new ShopItem[maxItems];
	}

	public void put(Weapon item,int quantity)
	{
	   if (numItems<maxItems){
	      table[numItems] = new ShopItem(item,quantity);
	      numItems++;
	   }

	}

	public ShopItem get(String key)
	{
	   int location = 0; //gets location in table based on key

	   while (location <numItems && key.compareTo(table[location].item.weaponName) != 0)
	   {  // not empty and not item
	      location++;
	   }
	   if (location<numItems){
	      return table[location];
	   }
	   return null;
	}

	public void printTable()
	{
	   int count = 0;
	   for (int x = 0; x < numItems; x++)
	   {
	        System.out.println("Name: " +table[x].item.weaponName+"
Damage:"+table[x].item.damage+"    Cost:"+table[x].item.cost+"     Quantity in
stock:"+table[x].numberInStock);
	   }
	}
}
```

---------- Main class ----------

```
public static int getInteger(Scanner sc,String message){
        System.out.print(message);
        while (!sc.hasNextInt())
        {
           sc.nextLine(); //clear the invalid input ...
           System.out.print(message);
        }
        return sc.nextInt();
    }

    public static double getDouble(Scanner sc,String message){
        System.out.print(message);
        while (!sc.hasNextDouble())
        {
           sc.nextLine(); //clear the invalid input ...
           System.out.print(message);
        }
        return sc.nextDouble();
    }


    public static void addWeapons(ArrayManager h,Scanner sc)
    {
        System.out.println("**********WELCOME TO THE WEAPON ADDING MENU*********");
        String weaponName; int weaponRange; int weaponDamage; double weaponWeight; double
weaponCost;
        int quantity;
        System.out.print("Please enter the NAME of the Weapon ('end' to quit):");
        weaponName=sc.next();
        while (weaponName.compareTo("end") != 0)
        {
           weaponRange= getInteger(sc,"Please enter the Range of the Weapon (0-10):");
           weaponDamage=getInteger(sc,"Please enter the Damage of the Weapon:");
           weaponWeight= getDouble(sc,"Please enter the Weight of the Weapon (in pounds):");
           weaponCost=getDouble(sc,"Please enter the Cost of the Weapon:");
           Weapon w = new Weapon(weaponName, weaponRange, weaponDamage, weaponWeight,
weaponCost);
           quantity=getInteger(sc,"Please enter the quantity in stock:");
           h.put(w,quantity);
           System.out.print("Please enter the NAME of another Weapon ('end' to quit):");
           weaponName = sc.next();
        }
    }
```

```
public static void showRoomMenu(ArrayManager ht,Player p){
   System.out.println("WELCOME TO THE SHOWROOM!!!!");
   ht.printTable();
   System.out.println("You have "+p.money+" money.");
   System.out.println("Please select a weapon to buy('end' to quit):");
}

public static void showRoom(ArrayManager ht, Player p,Scanner sc)
{
   String choice;
   showRoomMenu(ht,p);
   choice=sc.next();
   while (choice.compareTo("end") != 0 && !p.inventoryFull())
   {
      ShopItem si = ht.get(choice);
      if (si != null)
      {

            p.buy(si.item);
            p.withdraw(si.item.cost);
            si.numberInStock--;

      }
      else
      {
         System.out.println(" ** "+choice+" not found!! **" );
      }
      showRoomMenu(ht,p);
      choice = sc.next();
   }
   System.out.println("");
}

public static void main(String[] args)
{
   Scanner sc = new Scanner(System.in);
   String pname;
   System.out.println("Please enter Player name:");
   pname=sc.next();
   Player pl= new Player(pname,45);
   ArrayManager ht= new ArrayManager(101);
   addWeapons(ht,sc);
   showRoom(ht, pl,sc);
   pl.printCharacter();

}
```