```java
package gameStore01;
import java.util.*;
public class GameStore01 {

	final String[] MANUFACTURER = {"Microsoft", "Sony", "Nintendo", "Sega"};
	final String[] MODEL = {"Xbox One S", "Xbox One X", "Switch", "2DS XL", "3DS",
"PlayStation 4", "PlayStation 4 Pro", "Dreamcast"};
	final String[] STORE_INVENTORY_PRODUCTS = {"Game", "Book", "Game Console",
"Game Accessory"};
	final String[] CONSOLE_ACCESSORIES = {"Keyboard", "VR Headset", "Earphones",
"Controller"};
	final int ILLEGAL_VALUE = Integer.MAX_VALUE;
	private final int ARRAY_SIZE = 5;

	public String getManufacturer (Scanner kb) {
		boolean success = false;
		int rtn = ILLEGAL_VALUE;
		//do until success
		do {
			System.out.println( "Select the manufacturer's number from the
list.");
			//prints a menu using any values in MANUFACTURER array
			for ( int i = 0; i < MANUFACTURER.length; i++ ) {
				System.out.println( i + ": " + MANUFACTURER[i]);
			}
			System.out.print("Manufacturer: ");
			//if there's a number, and it's between 0 and the number of items
in the menu, set rtn value to response value
			//else, the number is beyond the scope of the menu
			if (kb.hasNextInt()) {
				int response = kb.nextInt();
				if ( 0 <= response && response < MANUFACTURER.length) {
					success = true;
					rtn = response;
				} else {
					System.out.println( "The value of the response <" +
response +
							"> is either less than zero or greater
than " +
							(MANUFACTURER.length-1) + ".");
					System.out.println( "Please select a valid
number.");
				}
			} else {
				System.out.println( "The response \"" + kb.nextLine() +
"\" is not valid.");
				System.out.println( "Please select a valid number.");
				System.out.println();
			}
		} while (!success);
		return MANUFACTURER[rtn];
	} // getManufacturer()

	public String getModel( Scanner kb ) {
		boolean success = false;
```

```
		int rtn = ILLEGAL_VALUE;
		do {
			System.out.println( "Select the Model's number from the list.");
			for ( int i = 0; i < MODEL.length; i++ ) {
				System.out.println( i + ": " + MODEL[i]);
			}
			System.out.print("Model: ");
			if (kb.hasNextInt()) {
				int response = kb.nextInt();
				if ( 0 <= response && response < MODEL.length) {
					success = true;
					rtn = response;
				} else {
					System.out.println( "The value of the response <" +
response +
							"> is either less than zero or greater
than " +
							(MODEL.length-1) + ".");
					System.out.println( "Please select a valid
number.");
				}
			} else {
				System.out.println( "The response \"" + kb.nextLine() +
"\" is not valid.");
				System.out.println( "Please select a valid number.");
				System.out.println();
			}
		} while (!success);
		return MODEL[rtn];
	} // getModel()

	public String getType( Scanner kb ) {
		boolean success = false;
		int rtn = ILLEGAL_VALUE;
		do {
			System.out.println( "Select the type of accessory from the
list.");
			for ( int i = 0; i < CONSOLE_ACCESSORIES.length; i++ ) {
				System.out.println( i + ": " + CONSOLE_ACCESSORIES[i]);
			}
			System.out.print("Type: ");
			if (kb.hasNextInt()) {
				int response = kb.nextInt();
				if ( 0 <= response && response <
CONSOLE_ACCESSORIES.length) {
					success = true;
					rtn = response;
				} else {
					System.out.println( "The value of the response <" +
response +
							"> is either less than zero or greater
than " +
							(CONSOLE_ACCESSORIES.length-1) + ".");
					System.out.println( "Please select a valid
number.");
```

```
                        }
                    } else {
                        System.out.println( "The response \"" + kb.nextLine() +
"\" is not valid.");
                        System.out.println( "Please select a valid number.");
                        System.out.println();
                    }
                } while (!success);
                return CONSOLE_ACCESSORIES[rtn];
        } // getType()

        public GameConsole readGameConsole(Scanner kb) {
                //what we're entering
                System.out.println("Entering a Game Console");
                System.out.println();
                //prompt manufacturer
                System.out.println("Manufacturer");
                System.out.println("------------");
                String manufacturer = getManufacturer(kb);
                System.out.println("Entering a " + manufacturer + " console");
                System.out.println();
                //prompt model
                System.out.println("Model");
                System.out.println("-----");
                String model = getModel(kb);
                System.out.println("Entering a(n) " + model);
                System.out.println();
                //prompt warrantyPeriod
                System.out.print("Warranty Period (months): ");
                int warrantyPeriod = 0;
                boolean done = false;
                while (!done) {
                    if (kb.hasNextInt()) {
                        warrantyPeriod = kb.nextInt();
                        done = true;
                    } else {
                        System.out.println("Please enter a valid number");
                    }
                }
                //create GameConsole
                return new GameConsole(manufacturer, model, warrantyPeriod);
        } // readGameConsole()

        public Game readGame(Scanner kb) {
                //what we're entering
                System.out.println("Entering a Game");
                System.out.println();
                //prompt publisher
                System.out.print("Publisher: ");
                String publisher = kb.nextLine();
                //prompt title
                System.out.print("Title: ");
                String title = kb.nextLine();
                //prompt manufacturer
                System.out.println("For use with what manufacturer?");
```

```
		System.out.println();
		System.out.println("Manufacturer");
		System.out.println("------------");
		String manufacturer = getManufacturer(kb);
		System.out.println("For use with " + manufacturer);
		System.out.println();
		//prompt model
		System.out.println("For use with what model?");
		System.out.println();
		System.out.println("Model");
		System.out.println("-----");
		String model = getModel(kb);
		System.out.println("Entering a(n) " + model + " game");
		System.out.println();
		//create game
		return new Game(publisher, title, manufacturer, model);
	} // readGame()

	public Book readBook(Scanner kb) {
		//what we're entering
		System.out.println("Entering a Book");
		System.out.println();
		//prompt publisher
		System.out.print("Publisher: ");
		String publisher = kb.nextLine();
		//prompt title
		System.out.print("Title: ");
		String title = kb.nextLine();
		//prompt author
		System.out.print("Author: ");
		String author = kb.nextLine();
		//prompt copyrightYear
		System.out.print("Copyright Year: ");
		int copyrightYear = 0;
		boolean done = false;
		while (!done) {
			if (kb.hasNextInt()) {
				copyrightYear = kb.nextInt();
				done = true;
			} else {
				System.out.println("Please enter a valid number");
			}
		}
		//create Book
		return new Book(publisher, title, author, copyrightYear);
	} // readBook()

	public ConsoleAccessory readConsoleAccessory (Scanner kb) {
		//what we're entering
		System.out.println("Entering a game accessory");
		System.out.println();
		//prompt manufacturer
		System.out.println("For use with what manufacturer?");
		System.out.println();
		System.out.println("Manufacturer");
```

```
        System.out.println("------------");
        String manufacturer = getManufacturer(kb);
        System.out.println("For use with " + manufacturer);
        System.out.println();
        //prompt model
        System.out.println("For use with what model?");
        System.out.println();
        System.out.println("Model");
        System.out.println("-----");
        String model = getModel(kb);
        System.out.println("Entering a(n) " + model + " accessory");
        System.out.println();
        //prompt type
        System.out.println("What type of accessory are you entering?");
        System.out.println();
        System.out.println("Type");
        System.out.println("----");
        String type = getType(kb);
        System.out.println("Entering a " + type);
        System.out.println();
        //prompt warrantyPeriod
        System.out.print("Warranty Period (months): ");
        int warrantyPeriod = 0;
        boolean done = false;
        while (!done) {
              if (kb.hasNextInt()) {
                    warrantyPeriod = kb.nextInt();
                    done = true;
              } else {
                    System.out.println("Please enter a valid number");
              }
        }
        //create ConsoleAccessory
        return new ConsoleAccessory(manufacturer, model, type, warrantyPeriod);
} // readGameConsole()

public void run() {

        Book[] books = new Book[ARRAY_SIZE];
        ConsoleAccessory[] accessories = new ConsoleAccessory[ARRAY_SIZE];
        Game[] games = new Game[ARRAY_SIZE];
        GameConsole[] consoles = new GameConsole[ARRAY_SIZE];

        int nextBook = 0;
        int nextAcc = 0;
        int nextGame = 0;
        int nextConsole = 0;

        Scanner kb = new Scanner(System.in);

        System.out.println( "Game Store Inventory System");
        System.out.println( "--------------------------");
        System.out.println();

        boolean done = false;
```

```
            do {
                  System.out.println();
                  System.out.println( "What type of inventory item would you like
to enter?");
                  //print menu
                  for ( int i = 0; i < STORE_INVENTORY_PRODUCTS.length; i++ ) {
                        System.out.println( i + ": " +
STORE_INVENTORY_PRODUCTS[i]);
                  }

                  //make sure input is an integer
                  boolean gotNum = false;
                  int choice = Integer.MIN_VALUE;
                  String choiceStr = null;
                  do {
                        System.out.print( "Selection: ");
                        if ( kb.hasNextInt() ) {
                              choice = kb.nextInt();
      // Get the integer
                              kb.nextLine();
            // consume the end-of-line character

                              // Make sure the value is within the range of the
menu items.
                              if (  0 <= choice && choice <
STORE_INVENTORY_PRODUCTS.length  ) {
                                    choiceStr = STORE_INVENTORY_PRODUCTS[choice];
                                    gotNum = true;
            // flag that we have the number
                              } else {
                                    System.out.println( "Your selection " +
choice + " is not valid.");
                              }
                        } else {
                              String txt = kb.nextLine();
      // Not a number. Get what the user entered.
                              System.out.println( "\"" + txt + "\" is not a valid
number. Try again");
                        }
                  } while ( !gotNum );

                  //run result of choice
                  switch ( choiceStr ) {
                  case "Game":
                        if (nextGame < ARRAY_SIZE) {           // make sure
there's still room in the Game array
                              games[nextGame] = readGame(kb);  // There is, so put
the new game in.
                              nextGame++;                                    //
Increment the index so it points to the next

      // empty slot.

                        } else {
      // There's no room left.
```

```
                                System.out.println("Sorry, there's no room in the
array for another game");
                        }
                        break;
                case "Book":
                        if (nextBook < ARRAY_SIZE) {
                                books[nextBook] = readBook(kb);
                                nextBook++;
                        } else {
                                System.out.println("Sorry, there's no room in the
array for another book");
                        }
                        break;
                case "Game Console":
                        if (nextConsole < ARRAY_SIZE) {
                                consoles[nextConsole] = readGameConsole(kb);
                                nextConsole++;
                        } else {
                                System.out.println("Sorry, there's no room in the
array for another game console");
                        }
                        break;
                case "Game Accessory":
                        if (nextAcc < ARRAY_SIZE) {
                                accessories[nextAcc] = readConsoleAccessory(kb);
                                nextAcc++;
                        } else {
                                System.out.println("Sorry, there's no room in the
array for another console accessory");
                        }
                        break;
                default:
            // Just in case.
                        System.out.println( "*** Error *** You shouldn't get
here!");
                        System.exit(100);
      // Bail out.
                }

                System.out.print( "Do you want to do another? (Y/N): ");
                String ans = kb.next();

                // Just look at the first letter of what was entered on the
keyboard
                if ( !ans.toUpperCase().substring(0, 1).equals("Y")) {
                        done = true;
                }
            } while ( !done );
            //print books
            System.out.println();
            System.out.println("Printing Books");
            for ( int i = 0; i < nextBook; i++ ) {
                System.out.println( i + ": " + books[i].toString());
            }
            //print accessories
```

```
            System.out.println();
            System.out.println("Printing Console Accessories");
            for ( int i = 0; i < nextAcc; i++ ) {
                  System.out.println( i + ": " + accessories[i].toString());
            }
            //print games
            System.out.println();
            System.out.println("Printing Games");
            for ( int i = 0; i < nextGame; i++ ) {
                  System.out.println( i + ": " + games[i].toString());
            }
            //print consoles
            System.out.println();
            System.out.println("Printing Game Consoles");
            for ( int i = 0; i < nextConsole; i++ ) {
                  System.out.println( i + ": " + consoles[i].toString());
            }

            //done
            System.out.println( "All done!");
            kb.close();
      }//run()

      public static void main(String[] args) {

            // Declare and instantiate the class GameStore00
            GameStore01 gs01 = new GameStore01();

            // Call the .run() method to do the work of this class.
            gs01.run();

      } // main()
} //class GameStore01
```